

AD-A144 720

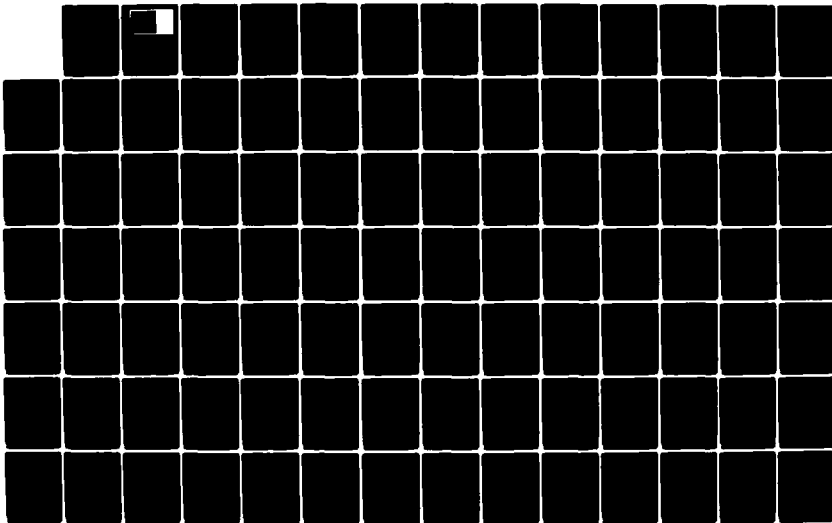
MICROSCOPE: A SOFTWARE SYSTEM FOR MULTIVARIATE ANALYSIS  
(U) WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER  
P ALFELD ET AL. JUN 84 MRC-TSR-2701 DAAG29-80-C-0041

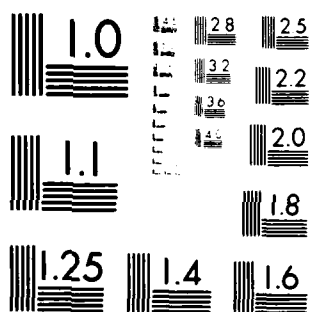
1/2

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A144 720

MRC Technical Summary Report #2701

MICROSCOPE: A SOFTWARE SYSTEM FOR  
MULTIVARIATE ANALYSIS

Peter Alfeld and Bill Harris

Mathematics Research Center  
University of Wisconsin—Madison  
610 Walnut Street  
Madison, Wisconsin 53705

June 1984

(Received April 27, 1984)

JUN 27 1984

Approved for public release  
Distribution unlimited

Sponsored by

U. S. Army Research Office  
P. O. Box 12211  
Research Triangle Park  
North Carolina 27709

Department of Energy  
Washington, D. C. 20545

84 08 24 04 6

DTIC FILE COPY

UNIVERSITY OF WISCONSIN-MADISON  
MATHEMATICS RESEARCH CENTER

MICROSCOPE: A SOFTWARE SYSTEM FOR MULTIVARIATE ANALYSIS

Peter Alfeld\* and Bill Harris\*

Technical Summary Report #2701  
June 1984

ABSTRACT

This manual describes MICROSCOPE: a portable FORTRAN software system for the analysis of multivariate functions. Given an interpolation or approximation scheme, it allows the following questions, among others, to be answered: Does the scheme interpolate? How often is it differentiable? What functions does it reproduce exactly? If the scheme is polynomial, what is its polynomial degree? Where is the smoothness of a function reduced? Where are the bugs in a FORTRAN implementation?

AMS (MOS) Subject Classification: 65D05

Key Words: Interpolation, Approximation, Software, Several Variables,  
Computer Aided Geometric Design

Work Unit Number 3 (Numerical Analysis and Scientific Computing)

---

\*Department of Mathematics, University of Utah, Salt Lake City, Utah 84112.

---

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.  
Supported in part by the Department of Energy under Contract No. DE-AC02-82ER12046 A000.

## SIGNIFICANCE AND EXPLANATION

In Computer Aided Geometric Design and other applications of Multivariate Interpolation and Approximation it is often necessary to build complicated computational procedures. These should have certain properties according to the mathematical theory, but the construction may have failed due to oversights or simple programming errors. This report describes an interactive portable FORTRAN software system, MICROSCOPE, that allows the following questions, among others, to be answered: Given an approximation or interpolation scheme, does the scheme interpolate? How often is the approximant differentiable? What functions does it reproduce exactly? The system also allows the identification of bugs in a FORTRAN implementation of an approximation scheme, often pinpointing the faulty portion of the code within narrow limits. MICROSCOPE is also useful in the corroboration of theoretical results, in guiding a theoretical analysis, and for classroom use. In addition to documenting the program package, this manual also contains a thorough discussion of numerical differentiation.

For ordering information contact the first author.



AC  
N

A-1

---

The responsibility for the wording and views expressed in this descriptive summary lies with MRC, and not with the authors of this report.

## List of Contents

0. About This Manual .....	6
1. Introduction .....	8
1.1 Multivariate Interpolation and Approximation .....	8
1.2 The Basic Approach .....	9
1.3 The Alphanumeric Display .....	13
1.4 Summary .....	19
2. Applications .....	21
2.1 Interpolation .....	22
2.2 Smoothness .....	27
2.3 Degree of Precision .....	28
2.4 Degree of a Polynomial .....	30
3. Using MICROSCOPE .....	33
3.1 The MCRSCP Routine .....	34
3.2 The Command Mode .....	36
3.3 A Discontinuity in the Second Derivative .....	38
4. A Detailed Description of Features and Commands .....	44
4.1 Round-Off Effects .....	44
4.2 Derivatives of Step Functions .....	45
4.3 The Window .....	49
4.4 Reference Table of Commands .....	51
4.5 Commands to Control the Display .....	54
4.6 Commands to Control the Point of Examination .....	58
4.7 Commands to Control the Direction of Investigation .....	59
4.8 Commands to Control Accuracy, the Window, and Round-Off Effects .....	60
4.9 Commands to Control Execution .....	62
4.10 Commands to Obtain On-Line Help .....	64
4.11 Commands to Keep a Record .....	65
4.12 Commands to Analyze Cross Derivatives .....	65
4.13 Programming MICROSCOPE .....	67
4.14 The PLOT command for obtaining a <PLOT79> Display .....	69
4.15 User Intervention .....	73
4.16 Default settings .....	74
5. Installation Guide .....	75
5.1 Description of the Package .....	75
5.2 A Key to Selecting the Appropriate Files .....	80
Appendix I: Numerical Differentiation .....	82
I.1 The Round-Off Threshold .....	85

1.2 Round-off versus Truncation Errors .....	87
1.3 Derivatives of Step Functions .....	89
Appendix II: Organization of <u>MICROSCOPE</u> .....	92
II.1 Organization of the Program Package .....	92
II.2 Sampling Scheme and Logic .....	95
II.3 Common Blocks .....	104
Appendix III: The Test Package .....	110
Appendix IV: Examples for Graphical (<PLOT79>) Displays .....	115
References .....	116
Acknowledgements .....	117
Index .....	118

# List of Tables and Figures

Figure 1 The Alphanumeric Display .....	14
Figure 2 Interpolation to Position at Zero.....	22
Figure 3 .....	23
Figure 4 Interpolation at $x = 2$ .....	24
Figure 5 Interpolation to the First Derivative .....	25
Figure 6 .....	26
Figure 7 Free End Condition .....	27
Figure 8 Smoothness at $x = 1$ .....	28
Figure 9 A Zero Error .....	29
Figure 10 Round-Off with a Bias .....	30
Figure 11 A Cubic Function .....	31
Figure 12 .....	32
Figure 13 .....	38
Figure 14 .....	39
Figure 15 .....	40
Figure 16 .....	41
Figure 17 .....	42
Figure 18 At the limits of resolution .....	43
Figure 19 Derivatives of a Step Function .....	46
Figure 20 Derivatives of a C Function .....	47
Figure 21 Derivatives of a C Function .....	47
Figure 22 Derivatives of a C Function .....	48
Figure 23 Derivatives of a C Function .....	48
Figure 24 Derivatives of a C Function .....	49
Figure 25 An Invisible graph .....	56
Figure 26 The Disentangled Graphs .....	57
Figure 27 An Extended Discontinuity .....	67
Figure 28 A $3/2$ times differentiable function .....	113
Figure 29 Another $3/2$ times differentiable function .....	114
Table: Efficiency of Window Widths .....	50
Quick Reference Table of Commands .....	52
Table of Discretization Controlling Commands .....	60
Table: Relative Efforts .....	62
Table 1: Constants for Differentiation Formulas .....	84
Table 2: Threshold Values of $h$ .....	86
Table 3: Optimum Values of $h$ .....	88
Table 4: The Superiority of Testing Higher Derivatives .....	91
Epitome Flowchart .....	94
Outline of SGAMMA .....	104
Table of Test Parameters .....	112

---



## 0. About This Manual

This manual constitutes the comprehensive description of MICROSCOPE, its operation, applications, foundations, and inner workings.

Section 1.1 describes what kind of problems can be addressed by MICROSCOPE. Section 1.2 describes the basic approach and is fundamental. Section 1.3 describes the alphanumerical screen display and is needed for understanding the examples given throughout this manual. It also introduces some terms and concepts and serves as a prelude to the capabilities of MICROSCOPE.

Section 2 gives examples that illustrate four major applications (interpolation, smoothness, degree of precision, and polynomial degree). These do of course not exhaust the potential of MICROSCOPE. In section 3, one other application (discovery or identification of a point where smoothness is reduced) is used as a vehicle for introducing the mechanics of using MICROSCOPE.

Section 4 gives a detailed description of all commands and features, organized by type of feature or task to be accomplished. In regular use, this will probably be the most frequently consulted section.

In reading sections 2 and 3, it is desirable that a working version of MICROSCOPE is available so that hands on experience can be gained. If this is not the case the installation guide (section 5) should be consulted first. That section also contains a detailed discussion of the portability features and possible variants of MICROSCOPE.

Appendices are included that describe some technical features in detail: the differentiation formulas, the organization of the program package, and the test package. There should also be an appendix IV containing pen plots corresponding to the cruder alphanumerical graphics in the body of the manual. However, that appendix is not machine legible, and is distributed separately.

The manual itself is in machine readable form so that it can be distributed together with the package and can be printed locally, and read on line using for example the search features of a text editor. It was generated by a MICROSCOPE program, with the text of the manual being "notes" and the example screen displays being computed by MICROSCOPE and automatically embedded into the text. (Actually, the MICROSCOPE program generated a data file that was then fed into the text formatting program DOCUMENT, see Beebe, 1980.) Machine legibility, however, accounts for the somewhat awkward notation in some places. For example, greek letters have been given abbreviations in terms of roman letters.

Definitions are denoted by writing the defined term in single

quotation marks. Ordinary quotations are in double quotation marks.

The examples in this manual are reproducible since the set of routines used for their generation is supplied with the package and can be used for test purposes. To ensure uniformity of results obtained on different machines, a facility has been added to simulate rounding, and all examples have been computed in 10 digits arithmetic. More precisely, if  $x$  is the quantity to be rounded to  $D$  digits, say, then  $x$  is replaced by

$$z := (1 + \text{eps}_1 \cdot 10^{-D}) \cdot x + \text{eps}_2 \cdot 10^{-D}$$

where  $\text{eps}_1$  and  $\text{eps}_2$  are random numbers between  $-1$  and  $+1$ . The addition of the  $\text{eps}_2$  term is not standard but appropriate in the present context, because in investigations with MICROSCOPE small numbers are often due to taking differences between very close large numbers, leading to a cancellation of significant digits. The larger the magnitude of  $x$ , the more insignificant the second term will become. If  $x$  is roughly 1 or larger (in magnitude) our rounding is essentially equivalent to the standard rounding (where  $\text{eps}_2 = 0$ ).

This manual will rarely be read sequentially from front to back. Therefore, key ideas and concepts have sometimes been restated if they are crucial in the given context. It is hoped that this redundancy will contribute to the usefulness of this document.

Any criticisms, comments and suggestions about MICROSCOPE are welcome and should be directed to Peter Alfeld, Department of Mathematics, University of Utah, Salt Lake City, Utah 84112, 801-581-6842. We are particularly interested in any applications of MICROSCOPE that are not described in this manual.

## 1. Introduction

### 1.1 Multivariate Interpolation and Approximation

This manual describes a FORTRAN software package, MICROSCOPE, whose main applications are in the analysis of functions of several variables (or 'surfaces') as they occur e.g. in the approximation of multivariate data or in the design of geometric objects such as the body of an automobile or an aircraft. This is a currently very active research area with many unsolved and difficult problems. For a survey see Barnhill, 1983, and the references quoted therein.

MICROSCOPE is portable to the extent that it has passed the PFORT verifier (Ryder, 1974) with the exception that it uses some non-FORTRAN characters (namely ?,<,>,!,:,:). Some non-portable optional features are supported by variants of the code. These include the use of lower case letters in Input and Output, and the use of modifications of an alphanumerical display on a CRT terminal. If the software (<PLOT79>, Beebe 1979) and hardware are available more sophisticated and pleasing graphical displays can also be obtained. However, even if none of the additional features are provided by a particular installation, a working, if somewhat crude, version of MICROSCOPE can be constructed. See the installation guide (section 5) for details.

The objects that can be examined by MICROSCOPE are functions of from one to three independent variables. Allowing only up to three variables enabled us to contain all numerical information on the CRT screen and proved sufficient for all cases we have encountered. If a function of more than three variables must be examined MICROSCOPE can be applied to a suitable restriction of the function to a subdomain of dimension at most three. If there is sufficient interest, future versions of MICROSCOPE capable of handling more than three variables will be provided.

The functions of interest are usually obtained by applying an interpolation or approximation scheme to discrete or transfinite (i.e. infinitely many) data. As far as the use of MICROSCOPE is concerned it matters little if a function is obtained by interpolation or approximation (except that one will not need to test for interpolation in the latter case), so we will use the two terms interchangeably.

We will refer to the function that is being examined as the 'trial function'. The trial function may be different from the interpolating or approximating function in which the final user of a numerical scheme is interested. For example, it is often useful to generate data from a 'primitive function' and then use the difference between the primitive function and its interpolant as the trial function.

This makes discontinuities in derivatives large relative to the derivative value and thereby amplifies discontinuities which are then more easily detected.

Four types of questions occur most frequently in the analysis of the trial function:

-1- Does the scheme interpolate?

-2- How often is the trial function differentiable? (i.e. what is its 'degree of smoothness'?)

Note: Throughout almost all of this manual we will consider only functions that possess a step discontinuity in some derivative, for this is the type of smoothness limitation that occurs in multivariate interpolation and approximation. There are of course other

possibilities. For example, the function  $f(x) = x^{4/3}$  is once differentiable but its second derivative exhibits a pole rather than a step. The test package (see appendix III) can be used to gain some experience with such functions. In particular, the examples in appendix III show how to identify and distinguish cusps and poles.

-3- Which functions are reproduced exactly by the scheme? In particular, which is the maximum degree up to which all polynomials are reproduced exactly? In other words, what is the 'degree of (polynomial) precision' of the scheme?

-4- If the trial function (or some of its derivatives) are polynomial what is their degree?

It should be obvious that only in the very simplest of cases answers to these questions can be obtained from a plain display of the surface of interest. Even given infinite display accuracy (although of course any physical display is ultimately piecewise linear or even discrete), it seems impossible to distinguish optically a once differentiable function from one that is twice differentiable.

## 1.2 The Basic Approach

In the design of an approximation scheme one usually knows the 'critical sets' where the phenomena of interest such as discontinuities in derivatives, or interpolation to certain data, may occur. For example, in a bivariate interpolation scheme defined on a triangulation, critical sets are usually edges of triangles, possibly internal edges, and vertices of triangles. In a trivariate scheme defined on a tessellation into tetrahedra, critical sets are faces, edges, maybe internal faces and edges,

and internal and external vertices. If the critical set is not known it can be discovered with MICROSCOPE (see section 3), but for the present we will assume it is known.

The basic idea of MICROSCOPE is very simple: Pick a random point (the 'point of examination') in a specific critical set. We will refer to that set as 'the front'. Then determine a 'direction of investigation' and consider a line through the point of examination in the direction of investigation. This is the 'line of investigation'. Depending upon the application, the line of investigation may be contained in the front, or it may be at an angle (not necessarily a right one) to it. We will call any derivative in the direction of investigation a 'tangential derivative'. Any pure derivative in a direction other than the direction of investigation is a 'cross derivative' and its direction is the 'cross direction'. MICROSCOPE numerically approximates the relevant derivatives (pure or mixed) and displays the approximations on the CRT terminal. Depending on the resulting picture, one may then take further action, e.g. pick another point of examination or another direction of investigation, consider other derivatives, or improve the existing plot by altering the underlying numerical parameters.

In using MICROSCOPE it is important to choose the underlying domain and the primitive function so as to avoid artifacts due to the regularity of the domain or peculiarities of the primitive function, and to pick the points of examination randomly or arbitrarily within the critical sets. Usually different types of critical sets will be present and of course one will need to consider at least one representative of each set.

Numerical differentiation is a notoriously ill-conditioned numerical process. However, in MICROSCOPE, the appearance of round-off errors is immediately apparent because the points on the graph of the relevant derivative are scattered across the screen in a random fashion. Moreover, by plotting derivatives of a high degree with a large discretization parameter the limitations imposed by round-off errors can usually be overcome. For example, if the expected discontinuity in a specific derivative is too unpronounced to be visible at the large value of the discretization parameter necessitated by round-off errors, then higher order derivatives can be computed (on an even coarser discretization). These will be approximations of the Dirac Delta function and its derivatives, and usually will pinpoint even weak discontinuities.

Numerically, we proceed as follows. Let  $F$  denote the trial function,  $P$  the point of examination, and  $d$  the direction of investigation. MICROSCOPE displays derivatives of  $F$  in the direction of  $d$  along the line of investigation with  $P$  being the center point. A built-in facility allows the replacement of  $F$  with a derivative of  $F$  in some other, independent, direction. More complicated mixed partial derivatives can be built by modifying the trial function, and MICROSCOPE offers some help with that as well. However, for the present we will restrict our attention to a

tangential derivative of  $F$ .

The directional derivatives are approximated at points

$$P(i) = P + i \cdot s \cdot d, \text{ where } i = -N, -N+1, \dots, N-1, N$$

and the quantity  $s > 0$  determines the spacing of the points. It will be referred to as the 'display interval' or 'interval' for short. The number  $N$  determines the resolution of the display (as well as the numerical effort). At each point  $P(i)$ , we approximate the  $k$ -th derivative of  $F$  in the direction  $d$  at the  $i$ -th point by a quantity  $Q_i^{(k)}$ .

The range of  $k$  in the present version of MICROSCOPE is from zero to six. We have found this sufficient for all of our applications. The quantities  $Q_i^{(k)}$  are determined by numerical differentiation (i.e. by differentiating an interpolating polynomial). The formulas, given below, were determined by the following criteria:

- 1- Central Differences should be employed in order to avoid artifact due to asymmetry.
- 2- The discretization parameter (which does not necessarily equal  $s$ ) should be such that the number of points affected by the suspected discontinuity (in some derivative) is independent of the order of the derivative.
- 3- Other than the above, the degree of the interpolating polynomial should be as low as possible.

Furthermore, we picked discretization formulas that use points at equally spaced intervals. This may be changed in future versions of MICROSCOPE.

Denoting the 'discretization parameter' by  $h > 0$ , the above criteria give rise to the following formulas:

$$Q_i^{(0)} = F_i$$

$$Q_i^{(1)} = (-F_{i-1} + F_{i+1}) / (2 \cdot h)$$

$$Q_i^{(2)} = (F_1 - 2F_5 + F_9)/h^2$$

$$Q_i^{(3)} = 4 \cdot (-F_1 + 2F_3 - 2F_7 + F_9)/h^3$$

$$Q_i^{(4)} = 16 \cdot (F_1 - 4F_3 + 6F_5 - 4F_7 + F_9)/h^4$$

$$Q_i^{(5)} = 243 \cdot (-F_1 + 4F_2 - 5F_4 + 5F_6 - 4F_8 + F_9)/(2h^5)$$

$$Q_i^{(6)} = 729 \cdot (F_1 - 6F_2 + 15F_4 - 20F_5 + 15F_6 - 6F_8 + F_9)/h^6$$

where

$$F_1 = F(P(i) - hd)$$

$$F_2 = F(P(i) - \frac{2hd}{3})$$

$$F_3 = F(P(i) - \frac{hd}{2})$$

$$F_4 = F(P(i) - \frac{hd}{3})$$

$$F_5 = F(P(i))$$

$$F_6 = F(P(i) + \frac{hd}{3})$$

$$F_7 = F(P(i) + \frac{hd}{2})$$

$$F_8 = F(P(i) + \frac{2hd}{3})$$

$$F_9 = F(P(i) + hd)$$

The first four of the above formulas can be found in Abramowitz and Stegun, 1968, p. 914, and the last two can be derived by the techniques described there. Higher order formulas can of course also be derived.

(k)

It is interesting to note that  $Q_i$  considered as a function of  $x_i$  (or a function of a continuously varying parameter  $i$ ) is a function of the same degree of smoothness as the trial function itself. Just the number of points of reduced smoothness is increased, there being one corresponding to each point in the differentiation formula passing through the point of reduced smoothness. Thus numerical differentiation can be thought of as a smoothing process. (This contrasts starkly with its numerical property of a roughing procedure due to round-off effects.) The figures in section 4.2 illustrate the smoothing properties of numerical differentiation.

For a more detailed discussion of the truncation and round-off properties of these formulas see Appendix I.

### 1.3 The Alphanumerical Display

In this subsection, the alphanumerical screen display generated by MICROSCOPE is described in detail. The purpose of the description is twofold: First, it is necessary for an understanding of the examples sprinkled throughout this manual. Second, however, it also serves as an illustration of some of the capabilities of MICROSCOPE and as a first introduction to the power and versatility of the package.

During a MICROSCOPE session the CRT screen will usually be occupied by the 'alphanumerical display' like the one illustrated in figure 1. (More sophisticated graphical displays can also be obtained, see sections 4.14 and 5.)



```

222      I      :      .....
. 22      I      :      * I      22 .
. 222     I      :      * I      222 .
. 22      I      :      I      22 .
. 22      I      :      I      22 .
. 222     I      :      * I      222 .
. 22      I      :      I      22 .
. 222     I      :      I      222 .
7654321..876543212287654321.987654321+123456789.123456782212345678..1234567
      222     I      :      I      222
      22      I      :      * I      22
      22      I      :      I      22
      222     I      :      I      222
      221     *      :      122
      22 *      :      22
.....2222222.....

```

---

CD: deg = 1 dir = ( 0.000000D+00, 1.000000D+00) ch = 6.0000D-04  
Point = ( 0.000000D+00, 1.000000D+00) s = 5.0000D-03  
Direction = ( 1.000000D+00, 0.000000D+00) h = 3.0000D-02  
F0 ( 9.39D-09, 1.27D-02) F2 ( 1.20D-01, 2.22D+00) F3 (-1.20D+01, 1.21D+01)  
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 174

Figure 1 Demonstration of the Alphanumerical Display

In this example, the trial function is  $f(x,y) = x^2 y \sin(x+y)$ . We analyze a cross derivative of it in the direction (0,1). The direction of investigation is (1,0), and the point of examination is (0,1). Because of the absolute value term, the cross derivative is twice but not three times tangentially differentiable at the point of examination.

We explain the display starting at the top and referring to lines either by their position or by the first word contained in them.

The display is naturally divided into two parts: The first 15 lines constitute the 'graphical display', and lines 17 through 21 form the 'numerical display'. Line 16 separates the two displays and is present only if there is sufficient space available.

#### -1- The Graphical Display

Every MICROSCOPE display shows the graphs of a set of tangential derivatives of degrees between 0 and 6. The function that is being differentiated tangentially is called the 'display function'. It may be the trial function itself (the most frequently occurring case), or, as in this example, a pure cross derivative of a degree between 1 and 6.

Distance along the horizontal axis corresponds to displacement in the direction of investigation from the point of examination (which corresponds to the center of the horizontal axis). The scale on that axis has the same meaning for all graphs shown. The vertical axis corresponds to function values, but the vertical scale is different for each derivative, and is chosen such that the graph fills the entire vertical extent of the graphical display. This normalization is carried out and modified automatically as the investigation proceeds. A period denotes the graph of the display function, and the digit  $k$ , say, where  $k$  is between 1 and 6 and denotes the  $k$ -th tangential derivative. In this example, the display function and its second and third tangential derivatives are being shown.

Sometimes some derivative is more interesting than others, and then it may be useful to accentuate it with asterisks. In the above example, the third derivative has been so emphasized. Usually, the graph of a given derivative overwrites the graph of any lower order derivative. However, the graph of an accentuated derivative overwrites all other graphs.

Both the number of lines and the number of columns in the display can be set according to specific purposes. Throughout this manual we will use 75 columns and 15 rows for the graphical display as this is appropriate for the width of the printed output. A width of 79 might be preferable for a standard CRT terminal displaying 80 columns; 135 columns and a larger number of rows might be used for printed output; and a smaller number is sometimes useful for preliminary investigations when the trial function is expensive to evaluate. An even number of columns is normally undesirable because then the point of examination does not lie in the precise center of the display.

Line 8 in the example contains a horizontal scale counting from the center to the left and right. This will usually be omitted so as not to clutter the display, but it is sometimes useful for the identification of points of interest. The distance from the center can be used to inquire about numerical values of certain derivatives, or to shift the graph left or right as required by the circumstances. The scale can be turned on or off, or replaced with a line of hyphens. The graph of any function overwrites the horizontal scale. The center of the display can optionally be marked with a "+" sign. This is sometimes useful to pinpoint symmetry properties. In the above example the graph of the third derivative passes through the center of the display. The center mark overwrites the graph of any function passing through it.

The central column of colons marks the column corresponding to the point of examination. The two columns of 1's outline 'the window', i.e. that set of points at which the numerical differentiation is affected by the behavior of the trial function at the point of examination. (This is not always quite true when cross derivatives are being examined, see the

discussion in section 4.11). For every point in the window, the interval about the point spanned by the differentiation stencil contains the point of examination. The differentiation formulas are chosen such that the window is identical for all tangential derivatives. Obviously, for the display function itself the window contains only the point of examination. For its derivatives, the window contains all points that can be written as  $P + td$ , where  $\text{abs}(t)$  does not exceed the value of  $h/s$ , (and, as before,  $P$  is the point of examination,  $d$  is the direction of investigation,  $h$  is the discretization parameter, and  $s$  is the interval). We define  $w := 2h/s$  to be the 'window width'. Obviously, by picking the window width  $< 1$ , it is possible to generate displays in which the only point contained in the window is  $P$  itself. However, for several reasons it is usually preferable to use a larger window width. This aspect is discussed in more detail in section 4.1. The default value of the window width is 12, as in the example, but this can be changed as needed.

Without examining the numerical display, it is apparent from the graphical display that some function is being investigated that is twice but not three times differentiable since the (accentuated) third derivative shows a clear step discontinuity. Note how that step is smeared out over the window.

#### -2- The numerical display

The first line of the numerical display starting with "CD:" indicates that the derivatives are mixed partials, i.e. they are tangential derivatives of a cross derivative (of the trial function). This line is absent if the trial function itself is being examined, and the line carries only information that is pertinent to the cross derivative. In the present example, the degree of the cross derivative ("deg") is 1, the cross direction is (0,1), and the discretization parameter used for computing the derivative (denoted by "ch" corresponding to "h" for the tangential derivative) is 6D-4. The formulas for computing cross derivatives are equivalent to those for computing tangential derivatives.

The line starting with "Point" describes the point of examination (which is (0,1)) and the interval (which is 5D-3). This display implies that the trial function is bivariate. Appropriate modifications take place for functions of 1 or 3 variables. The next line starting with "direction" describes the direction of investigation (which is (1,0)) and the discretization parameter (which is 3D-2). Notice that the window width is  $2h/s = 12$  which is consistent with the graphical display

The "Point" and "Direction" descriptions are always present.

The next line, starting with "F0", gives the ranges of the displayed derivatives. Ranges are labeled  $F_k$  where  $k$  is the order of the tangential derivative. For example,  $F_2$  corresponds to the range of the second (tangential) derivative (of a first order cross derivative) which

here is (0.12.2.1). If more functions (up to seven) are displayed, then their ranges are also given, adding up to two lines to the numerical display as necessary.

The next and last line, starting with "I/O", contains information about the present configuration of MICROSCOPE.

The 10 integers following "I/O" are the FORTRAN channel numbers as they are employed by MICROSCOPE at the display time. They correspond to the following:

-1- 'Input Device' (here 25), i.e. the device from which commands are read. In an interactive session, this will usually be the terminal. But it may also be a file of MICROSCOPE commands that generate a specific setup, or a documentation like this manual.

-2- 'Output Device' (here 6), i.e. the device receiving the command prompts and any help information. Usually, this will also be the terminal, or, if MICROSCOPE is being driven by commands in a file, it may be a Null or Dummy device.

-3- 'Graphics Device' (here also 6), i.e. the device receiving the alphanumerical display illustrated in figure 1. This will again usually be the terminal, but it may also be a separate display device or a file receiving information for subsequent printing.

-4- 'Help Device' (here 1), i.e. the device containing the MICROSCOPE help file, which is supplied with the package. MICROSCOPE has several interactive help facilities and the information needed for running them is read out of this file when MICROSCOPE is first called. Usually you will not be concerned with precisely where MICROSCOPE receives its help information. However, the channel number is printed in the display to help prevent an accidental overwrite of the help file.

-5- 'Recording Device' (here 2), a device that can receive selected information from the MICROSCOPE session, e.g. copies of the screen display or 'notes' that explain the displays. This manual is the result of reading a set of instructions, i.e. a MICROSCOPE 'program' from the Input Device (a data file), processing it and writing the results onto the Recording Device.

-6- 'Restart Device' (here 3). At any time during a session, the current configuration of MICROSCOPE can be saved, allowing a later restart from the current status of the investigation. This is accomplished by writing the values of all relevant variables into a file pointed at by the Restart device.

-7- 'Logging Device' (here 27). It is possible to keep a diary of the MICROSCOPE session listing the commands and data entered as well as the results effected by them. This feature is activated by setting the Logging Device number to a non-zero value.

-8- 'Point Loading Device' (here 28). A device allowing you to read points from a file, rather than having to type them in.

-9- 'Direction Loading Device' (here 29). Similar to -8- for loading the direction of investigation.

-10- 'Cross Direction Loading Device' (here 30). Similar to -8- for loading the cross direction.

The first six I/O channel numbers are parameters of the master MICROSCOPE routine (see section 3.1) and can each be changed during a session. The last four channels are zero (i.e. unused) by default, and can be set to non-zero (i.e. active) values during a MICROSCOPE session. It is of course unnecessary to have all channel numbers distinct. For example, all three loading devices might be identical, allowing for the possibility of having just one list of points, directions, and cross directions.

Following the device numbers is a flag "NRML on" or "NRML off" (here on), indicating whether the direction of investigation entered into MICROSCOPE has been normalized or not. The "Direction" line always gives the direction as entered by the user. If NRML is off, the direction of investigation, d, is identical to that printed in the display, otherwise it is that printed in the display normalized to have unit Euclidean length. With the normalization on, tangential derivatives are the standard directional derivatives as they are defined in Calculus Textbooks. With the normalization off, they are the more versatile Gateaux derivatives as they are used frequently in multivariate interpolation and approximation. The NRML flag applies similarly to the cross direction.

Following the NRML flag is the word "current". This indicates that the numerical display corresponds to the graphics display. To explain why this might not be so requires a brief excursion into the Organization of MICROSCOPE. When drastic parameter changes are taking place it is often more efficient to accumulate them before generating the new display. Usually in that case the alphanumerical display is scrolled while commands are being given, and vanishes from the screen. However, sometimes it is preferable to view the old display rather than the commands given so far. In that case, the display can be recalled on the screen, and the word "current" is replaced by "GO pndg" (i.e. the GO command needs to be given to incorporate into the graphical display changes of parameters that may already be shown in the numerical display).

The last item in the last line is "CALLS = 174" where 174 is the number of evaluations of the trial function expended up to this point in the MICROSCOPE session. This allows monitoring the computational effort spent. A typical multivariate surface will be expensive to evaluate, and its cost will usually dominate the cost of running MICROSCOPE and thereby determine how long you have to wait for a response from the terminal. Therefore, a significant effort has been made to keep the number of necessary function evaluations small. This feature contributes in no small measure to the complexity of the MICROSCOPE source code. The example display has 75 columns and we are examining a first order cross derivative of the trial function. Each evaluation of the cross derivative requires two evaluations of the trial function. Displaying the graph of the cross derivative alone would therefore require a value of CALLS equal to  $2 \times 75 = 150$ . Note that computing and displaying the second and third tangential derivatives of the cross derivative requires an additional effort of only 16%. This is not a trivial accomplishment as may be suggested by the fact that the same display with the slightly different window width of 10 would require 330 evaluations. For a detailed discussion of the numerical effort associated with various window widths see section 4.3.

Notice that in the above example we identified a step discontinuity in a mixed partial derivative of fourth order (a third order tangential derivative of a first order cross derivative) without fuss or doubt. If this was a real investigation we could state now that the underlying scheme in any case is at most three times differentiable.

#### 1.4 Summary

With MICROSCOPE, one answers questions like those in section 1.1 for specific examples. Thus only negative results (e.g. that a scheme is not differentiable) can be proved strictly. However, positive results can also be obtained. If the examples are chosen carefully to exclude artifacts and make them representative, our program allows for answers to the above questions with a degree of confidence that borders on certainty! Naturally, a computational approach does not replace a rigorous proof. However, in the authors experience it is often possible to confirm (or shatter) fuzzy notions with MICROSCOPE, and to generate further insight into the phenomena under consideration. Encouraging MICROSCOPE results may help in building up the stamina needed for attempting a rigorous proof, and detailed investigations may help in providing ideas for carrying out the analysis.

Often, of course, MICROSCOPE will be employed in hindsight, when the analyst already knows the answers to questions like the above from theoretical reasoning. The examination with MICROSCOPE then serves to corroborate the reasoning, and also, more mundanely but no less importantly, to eliminate bugs in the implementation.

In the authors experience, MICROSCOPE proved invaluable as a debugging tool, often pinpointing within narrow limits that part of a code that was at fault, but also sometimes uncovering flaws in the theoretical approach, and, on a few exhilarating occasions, leading to the discovery of new and unlooked for results that could then be proved theoretically. Thus MICROSCOPE is useful in the identification as well as in the verification of results.

In the development of MICROSCOPE we were occasionally confronted with the objection that our alphanumerical displays are rather crude. Our primary response to this is that the very purpose of MICROSCOPE consists of translating very subtle properties into displays that make them glaringly apparent. A large part of our interest centers around step functions, and it seems fair to display them as such. Indeed, any curve fitting might disguise the very features in which we are interested. However, we do provide an interface to a sophisticated graphics package. See sections 4.14 and 5 for more information.

## 2. Applications

In this section, we illustrate the main applications of MICROSCOPE. Consider the simple example of interpolating to a primitive function  $p$  by a cubic spline  $S$ , say, according to the conditions:

$$\begin{aligned} S(x) &= a_0 + a_1x + a_2x^2 + a_3x^3 & \text{if } x < 1 \\ \text{and} \\ S(x) &= b_0 + b_1x + b_2x^2 + b_3x^3 & \text{otherwise} \end{aligned}$$

where the coefficients are defined by

$$S(0) = p(0), S(1) = p(1), S(2) = p(2), S'(2) = p'(2), S''(0) = 0$$

and  $S$  is twice differentiable at  $x = 1$ . Thus we require interpolation to 'position' (i.e. function values) at 0,1,2, and interpolation to the first derivative at 2 by a piecewise cubic function that is twice continuously differentiable. At 0 we impose additionally a "free end" condition. (There is an extensive literature on Splines, see e.g. de Boor, 1978, or Schumaker, 1981)

For our present purposes we assume that  $p(x)$  is the exponential, i.e.

$$p(x) = e^x$$

where  $e$  is the base of the natural logarithm. This function appears to be sufficiently non-polynomial to avoid the introduction of artifacts.

A simple calculation shows that

$$\begin{aligned} a_0 &= 1 \\ a_1 &= (-2e^2 + 12e - 9)/7 \\ a_2 &= 0 \\ a_3 &= (2e^2 - 5e + 2)/7 \\ \text{and} \\ b_0 &= (5e^2 - 16e + 12)/7 \\ b_1 &= (-17e^2 + 60e - 24)/7 \end{aligned}$$



$$b2 = \frac{(15e-48e+15)}{7}$$

$$b3 = \frac{(-3e+11e-3)}{7}$$

In the following examples we will verify that S with the above coefficients does indeed possess the required properties.

## 2.1 Interpolation

The basic idea of verifying interpolation is to plot the difference between the primitive function and its interpolant, and to check that the error and its appropriate derivatives are indeed zero. So we incorporate the exponential and S as defined above into MICROSCOPE (see section 3.1 for details), and start an interactive session. After giving the appropriate commands we obtain the following picture illustrating the situation at  $x = 0$ :

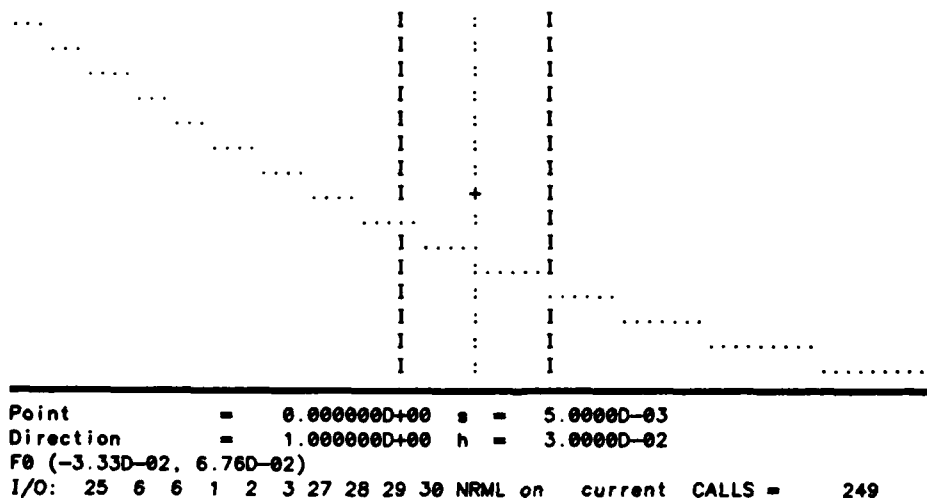


Figure 2 Interpolation to Position at Zero

We notice that the range of the error is from  $-0.0333$  to  $0.0676$ . Since the graph of the error function passes through the vertical axis slightly below the center, this range is consistent with the error being zero, i.e. with interpolation. A more reliable confirmation can be obtained by asking for the value of the displayed function at the center (i.e. 0) of the display. This procedure, which is not shown here, yields a value of  $-9.4D-11$ , which is zero up to round-off errors. (If you try to

reproduce this example with your own version of MICROSCOPE you may get a slightly different result because of the random nature of round-off errors.) An alternative approach consists of halving the interval, and thereby decreasing the range covered by the graph. In the numerical display the range should shrink correspondingly, maintaining the value of zero in its interior. Halving the interval yields the display:

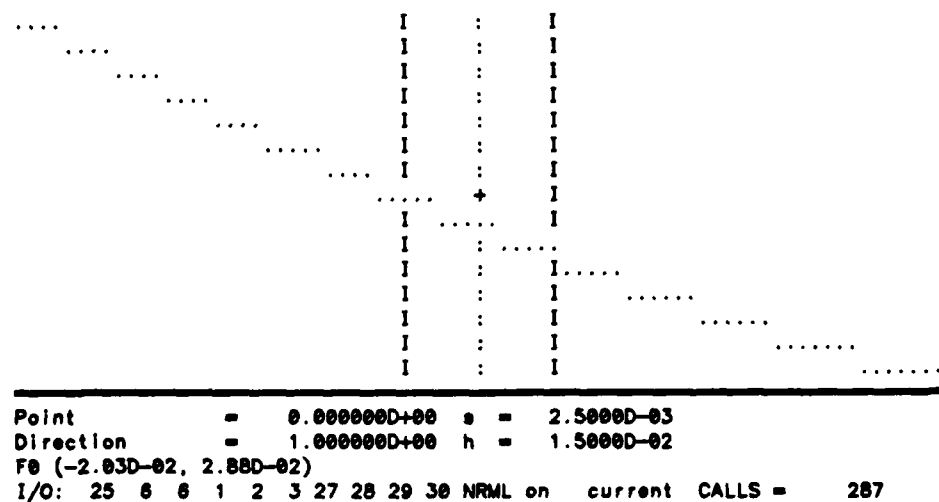
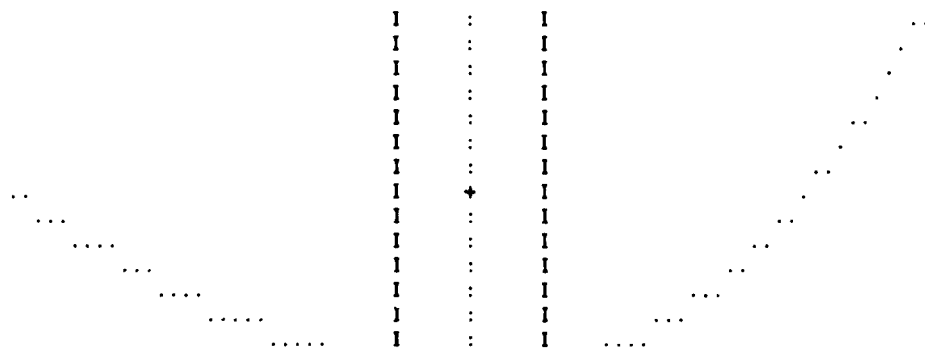


Figure 3

The range of the error has been reduced to from  $-0.0203$  to  $0.0266$  and a zero value of the error at the point of examination is still consistent with the graphical display. This procedure can be continued. Proceeding similarly at the point  $x = 1$  yields a similar result, and you might try this for yourself. At the point  $x = 2$ , however, we obtain the display:




---

```

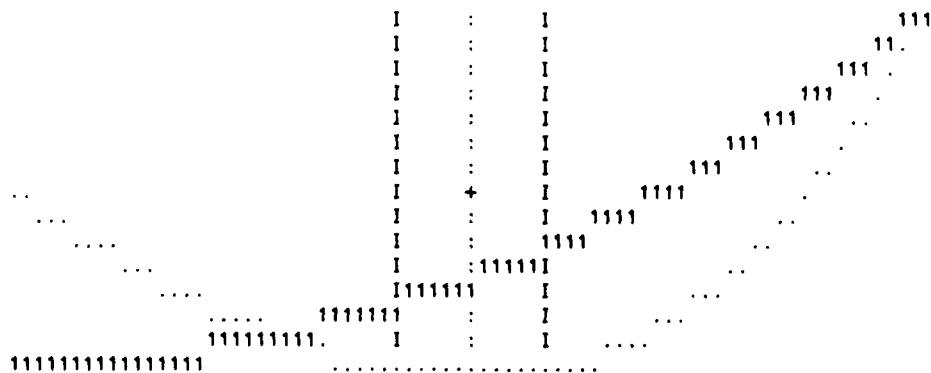
Point      = 2.000000D+00  s = 5.00000D-03
Direction  = 1.000000D+00  h = 3.00000D-02
F0 ( 2.64D-11, 1.42D-02)
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on  current CALLS = 362

```

---

Figure 4 Interpolation at  $x = 2$

The graph of the error function touches, but does not cross, the  $x$  axis, which is consistent with our interpolating to the first derivative as well as position. The value of the error at the center can be read directly from the numerical display and is  $-2.4D-11$  (you may get a slightly different value because of the random round-off errors). This confirms interpolation to position. Interpolation to the derivative is likely because the graph appears to have a horizontal tangent. However, confidence can be raised by computing the first derivative of the error as well, yielding the display:



Point = 2.000000D+00 s = 5.0000D-03  
 Direction = 1.000000D+00 h = 3.0000D-02  
 F0 ( 2.64D-11, 1.42D-02) F1 (-6.11D-02, 1.77D-01)  
 I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 374

Figure 5 Interpolation to the First Derivative

Inquiring about the value of the first derivative at the center yields the number 5D-4. This is much larger than the value for the error itself, and hence might cause some doubts. However, the derivative is calculated numerically and hence not exactly. The value reported by MICROSCOPE is actually the truncation error. This can be confirmed by halving the discretization parameter yielding the display:

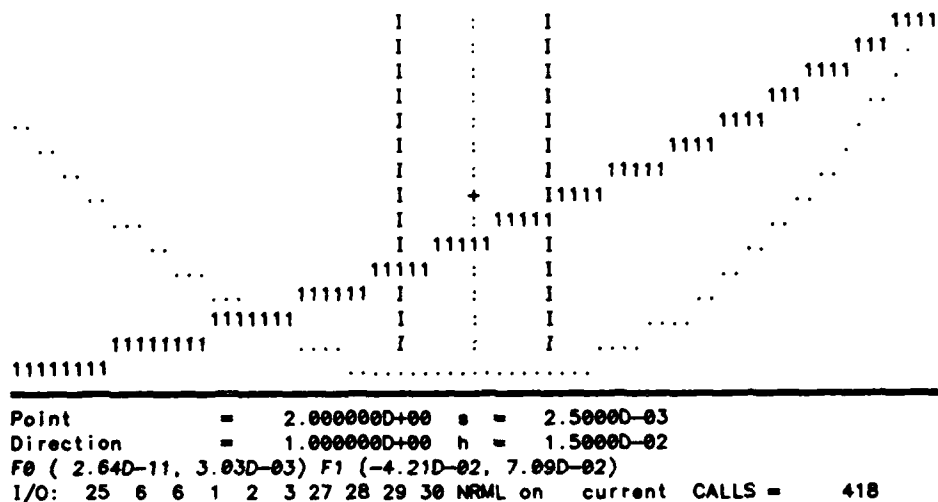


Figure 6

Now the value of the derivative is  $1.25D-4$  which is a reduction from the previous value by a factor 4. This is consistent with the assumption that the reported error actually is the truncation error since it decreases like the square of the discretization parameter  $h$  (see the discussion in Appendix I).

The free end condition ( $S''(2) = 0$ ) can be considered an interpolation condition. However, it cannot be checked by examining the error function, since the value of  $p''(2)$  is (in general) unknown. So one has to consider the interpolant itself. We obtain the display:



That the second derivative is indeed zero can now be verified using the same techniques as above (i.e. inquiring directly or using a decreased discretization parameter).

## 2.2 Smoothness

In the present example, the only critical set is the one containing the point  $x = 1$  (assuming we are confident that  $S$  has been programmed to be polynomial elsewhere). When investigating smoothness it is usually preferable to examine an error (if a primitive function is available) rather than the interpolant itself. In the present example, the error function at the single critical point yields the display:

```

33333333333333333333333333333333
I : 3 I .2222
I : 3 I 12222
I : I 112222
I : I .112222
I :3 I .1112222
I : 11122222
I +1112222
I111112222 I
.1111222223: I
1111222222I I
112222222 I : I
1222222 I 3 : I
1222222 I 3 : I
33333333333333333333333333333333 : I

```

---

Point = 1.000000D+00 s = 2.5000D-03  
Direction = 1.000000D+00 h = 1.5000D-02  
F0 ( 2.49D+00, 2.97D+00) F1 ( 2.39D+00, 2.90D+00) F2 ( 2.48D+00, 3.11D+00)  
F3 ( 2.73D+00, 4.06D+00)  
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 592

Figure 8 Smoothness at  $x = 1$ 

This display can be interpreted similarly as figure 1, and clearly shows a function that is twice but not three times differentiable. This is what we expect from our construction.

### 2.3 Degree of Precision

The 'precision class' of an approximation scheme is the set of functions that are reproduced exactly by the scheme. Of particular interest are the polynomials in the precision class. The largest number  $q$ , say, such that all polynomials of degree up to  $q$  are in the precision class is the degree of polynomial precision.

To decide if any particular function is in the precision class one approximates that function and investigates if the error is zero within round-off effects. This applies even to non-polynomial functions.

To answer the more narrow question of polynomial precision it is possible to proceed more systematically and to examine polynomials of increasing degree. The polynomials should be as general as possible, i.e. they should be unlikely to introduce misleading artifacts. One possibility is to choose the coefficients randomly. If really in doubt, one might examine each polynomial in a suitable basis of the appropriate polynomial space. The points of examination should be chosen such that each

appropriate part of the domain is covered. For instance, in the present example (which, incidentally, has linear precision) one should choose one point  $> 1$ , and one  $< 1$ , covering both cases in the definition of  $S$ .

The following figure illustrates the typical appearance of a zero error.

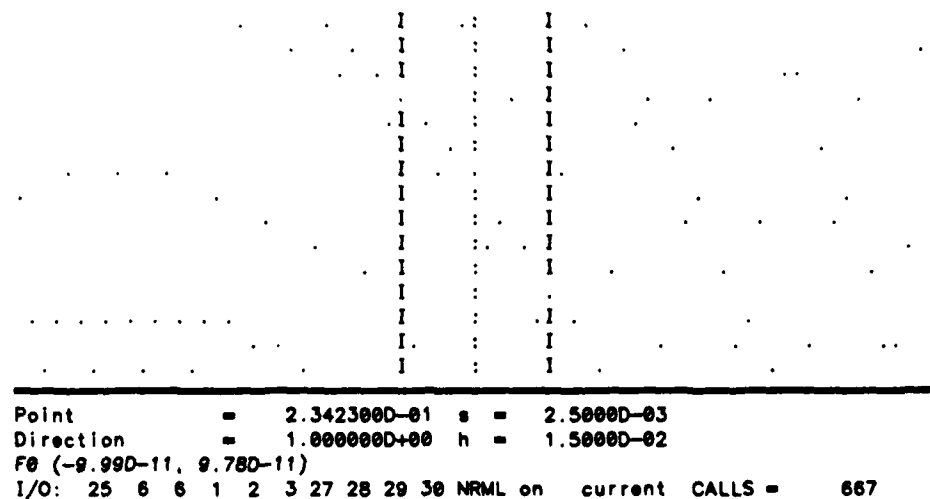
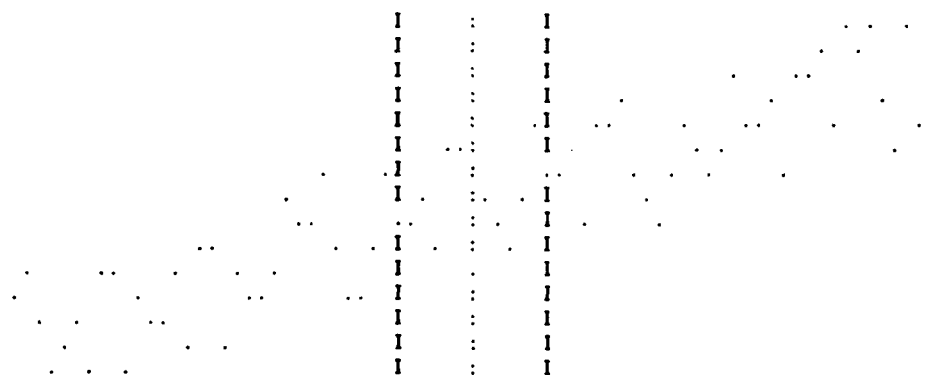


Figure 9 A Zero Error

Notice how the values of the function are spread randomly over the entire display, accurately pinpointing the round-off nature of the error. Sometimes, however, one has a situation in which the coefficients of the interpolant are evaluated inaccurately, leading to an error that has a scattered distribution with a bias like that in the following display:





Point = 2.3423000-01 s = 2.50000-03  
 Direction = 1.0000000+00 h = 1.50000-02  
 F0 ( 2.16D-10, 7.31D-10)  
 I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 742

Figure 10 Round-off With a Bias

#### 2.4 Degree of a Polynomial

Verifying that a certain function is a polynomial of a certain degree is often useful when examining cross derivatives. For example, in the bivariate Clough-Tocher scheme (Alfeld, 1984) which is piecewise cubic on a triangle, differentiability between triangles is forced by the requirement that the normal derivatives across edges be linear (rather than quadratic) along the edges. In this illustration, let us verify that our interpolant is indeed cubic. This can be accomplished by showing that the third derivative is constant or that the fourth derivative is zero (within round-off). One obtains for example the following display:

```

      3          I      I      22222
        3      3      I      I      222221
          3      I      I      3 3 3222211
            31      I      22222111
      3      3 3      I 3 : 3 313 222221111 3 3
        3      33 3 3 : 3 32322 1111 3 3
          3 3 3      3 3 I : 22222. 11113 3 3
      33      3 3 13 23222. 11113 3 3 3 3 3 3
        22222. 11111 I      3
      3      3 3      3 3 22222.111113 3 3 I      3
        22223. 11111 33 :3 I      3 3 3 3 3
      3      22223 311111 I : I      3 3 3
        2222211111 I : I      3 3 3
      222221111 I : I
2222211 3 I : I

```

Point = 2.342300D-01 s = 2.5000D-03  
 Direction = 1.000000D+00 h = 1.5000D-02  
 F0 ( 1.18D+00, 1.43D+00) F1 ( 1.29D+00, 1.41D+00) F2 ( 3.87D-01, 8.92D-01)  
 F3 ( 2.73D+00, 2.73D+00)  
 I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 841

Figure 11 A Cubic Function

Notice that the third derivative is clearly affected by round-off, without any apparent bias. On the other hand, since the displayed limits of its range are identical the size of the range is less than 1/273th of the derivative value. A closer examination reveals that the third derivative ranges from 2.7306 to 2.7323. A telltale sign of round-off errors is that they can be decreased even further by increasing the discretization parameter. Doubling h yields the display:

```

          I : I          3 322222
        3 3          I 3 : I          2222231
33          I : I          22222 111
          3 3 33          I : 333I          233223 111
        3 3 3          3 3 : I 3 22322. 31111
          3 3 3          I : I 22222. 111
        3 3 3          3 3 I 3 3332222.33 33 1111 3 3 3 3
        3 3          3 I 22222. I 1111 3 3 3
        3          33 3 32322. : 1111 3 3 3
          22222.I 3 : 1111333 3 3
        3 322222. I 11311 I 3 3
        3 3 22222. 11111 : I 3 3
          22222. 1111111 I : I
        23222. 111111111 I : I
2222211111111 3 I : I 3

```

```

Point      = 2.342300D-01 s = 5.00000D-03
Direction  = 1.000000D+00 h = 3.00000D-02
F0 ( 1.06D+00, 1.56D+00) F1 ( 1.27D+00, 1.50D+00) F2 ( 1.34D-01, 1.15D+00)
F3 ( 2.73D+00, 2.73D+00)
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 885

```

Figure 12

The range is now from 2.731361 to 2.731569 which is a reduction by a factor 8. This is consistent with the third derivative being constant since the round-off errors in the third derivative increase like the third power of h, see Appendix I. Further increases do not alter the basic scattered structure of the third derivative (until the range covered in the display includes the point  $x = 1$ , where the third derivative has a step discontinuity). Increasing the value of h should eventually decrease the round-off errors below a level at which the true change in the derivative becomes visible in the display if the third derivative was not constant. The fact that this does not happen indicates that there is no such change.

### 3. Using MICROSCOPE

This section contains a description of the master MICROSCOPE routine MCRSCP and a first introduction to the command language of MICROSCOPE. For a more detailed description consult section 4.

During an interactive session the terminal screen may be either 'active' or 'inactive'. On an active screen, changes in parameters are incorporated immediately into the alphanumeric display which is maintained on the screen. On an inactive screen, commands are accumulated and carried out only when the screen is activated by either the command GO or FORCE. An inactive screen shows the last few commands and the corresponding data whereas an active screen shows at most the current command and its data, in addition to the alphanumeric display. (However, an incomplete alphanumeric display may be called on an inactive screen, see the description of the command RSCREEN in section 4.5 and the discussion of the alphanumeric display in section 1.3.)

A second distinction applies to the way in which screen changes are implemented on an active screen. In one version of MICROSCOPE, which we call the 'screen version', the alphanumeric display is kept stationary and only modified rather than replaced. This version requires you or your system manager to provide four simple screen editing routines that are specific to each computer/terminal combination (see chapter 5.) If this facility is not available, or if you use a hardcopy terminal, then the 'scroll version' of MICROSCOPE may be employed, which scrolls the old screen and replaces it by a new version after each change.

The screen version is much preferable, and in this manual we usually assume it is available.

Another distinction pertains to the utilization of lower case letters. The use of lower case letters in a FORTRAN program is not portable, but now most installations do provide this facility. In this manual we assume lower case letters are available. If so, any lower case letters occurring in the input to the program are treated as if they were upper case. Thus commands can be given in lower case letters. Output from MICROSCOPE consists of a mixture of lower and upper case letters. However, to maintain portability we provide a version of MICROSCOPE that works solely with upper case letters. Section 5 contains more information.

### 3.1 The MCRSCP Routine

To put MICROSCOPE into operation, you will have to write a FORTRAN main program that calls the routine

```
MCRSCP(F, INPUT, OUTPUT, GRAPHIC, HELP, RECORD, RESTART,  
X      LINES, WIDTH, PLOT, NUMRCL, PROMPT)
```

The first parameter is the name of a DOUBLE PRECISION function (the trial function). It has the form

```
DOUBLE PRECISION FUNCTION F(X)  
DOUBLE PRECISION X  
DIMENSION X(1)
```

```
      .  
      .  
      .  
F = ...  
RETURN  
END
```

and must be declared EXTERNAL in the calling program. MICROSCOPE will set at most the first three components of X before calling F (i.e. it can handle only functions of up to three variables). Notice that F does not have any parameters other than the values of the independent variables. Usually, however, it will also depend on parameters describing e.g. the domain data structure. These are most conveniently passed to F in a common block.

All of the remaining parameters of MCRSCP are input integers. MCRSCP will not change them, and they may therefore be integer values rather than variables. The first six of them (i.e. INPUT, OUTPUT, GRAPHIC, HELP, RECORD, RESTART) are FORTRAN device numbers corresponding to the first six I/O numbers listed in the numerical display (see section 1.3). It is your responsibility to set up the necessary correspondences between device numbers and physical devices or disk files.

The last five parameters describe the physical dimensions of the screen display. Their meaning is as follows: (Numbers in parentheses give the value used for the examples in this manual, square brackets contain the allowable ranges for the parameters).

LINES (24): The total number of lines used on the terminal screen [ $10 < \text{LINES} < 58$ ].

WIDTH (75): The number of columns used for the graphical display [ $0 < \text{WIDTH} < 136$ ]. Usually one will choose as large a value as possible for maximum resolution. However, small values may sometimes be preferable for increased speed. For a symmetrically placed vertical axis, the value

of WIDTH should be odd. The numerical display occupies up to 75 columns. This is therefore the minimum physical width of the output device. WIDTH is also the number of points at which derivatives are sampled in the <PLOT79> display.

PLOT (15): The number of lines allocated to the graphical display [ $0 < \text{PLOT} < 49$ ]. Again, a value of PLOT as large as possible should be supplied. The numerical effort is independent of PLOT.

NUMRCL (8): The number of lines above the screen bottom at which the numerical display begins. [ $5 + \text{PROMPT} < \text{NUMRCL} < \text{LINES} - \text{PLOT}$ ].

PROMPT (2): The number of lines above the screen bottom at which the MICROSCOPE prompts are given. [ $1 < \text{PROMPT} < \text{LINES} - \text{PLOT} - 6$ ].

Usually, MCRSCP will be called only once during a session. However, it is possible to call your own routines from MCRSCP. You have to supply a subroutine SUBUSR without any parameters which can be called by MCRSCP. A detailed discussion of this procedure is given in section 4.15 on user intervention. For now we may assume that SUBUSR is a dummy routine

```
SUBROUTINE SUBUSR  
RETURN  
END
```

(which is supplied with the MICRO package and which you can of course write yourself).

When called, MCRSCP first prints an identifying stencil on the OUTPUT device, containing its version number and the last modification date. Next, it checks to see if the screen parameters satisfy the restrictions listed above. If not then MCRSCP exits, rather than terminates, in order to give you a chance to correct your mistake.

On the first call to MCRSCP it computes the machine round-off unit (see section 4.1) and prints its common logarithm to give you an idea of how many decimal digits are carried on the computer. The actual number relevant to your investigation may of course be less depending on the details of the arithmetic involved in evaluating the trial function.

Also on the first call only, MCRSCP then prints the first line of any available news (see section 4.10) to help you decide if you wish to read them.

After setting some defaults MCRSCP then enters the command mode, in which it accepts, interprets and carries out the commands entered on the Input Device.

### 3.2 The Command Mode

The MICROSCOPE prompt is of the form:

n >>

where n is the number of the command to be given. It starts at 1 and is incremented by 1 after each command.

A MICROSCOPE command is a string of at least two letters and digits. However, only the first two characters are significant and recognized by the parser. They must be entered in the first two columns immediately following the prompt. On an active screen, this will be in the same line as the prompt, on an inactive screen, or in the scrolling version, on the next line. Blank spaces are significant. (MICROSCOPE simply reads the command using a 2A1 format. The remaining columns can be filled with arbitrary characters. They are not read or processed by MICROSCOPE. This is sometimes useful for including comments in a program. A command line is terminated by pressing the RETURN button.

After a command you will be prompted for the appropriate data (if any are needed). The prompt describes what data are needed and is specific to the command. There are seven types of data requests:

- 1- No Data
- 2- A command name (to obtain help on the requested command)
- 3- 1 Integer
- 4- 2 Integers
- 5- 1 Double Precision variable
- 6- 1 Double Precision Vector
- 7- 2 Double Precision Vectors

The dimension of the vectors is identical to the number of variables of the trial function. (The default is 2, see section 4.18 for a listing of all defaults). Numbers are processed by an internal parser and are essentially format free. The following distinctions and restrictions apply, however:

- Blanks are ignored. For example "1 2" is considered the single integer 12.
- Numbers on the same line must be separated by commas.
- Numbers can also be separated by writing them on separate lines

(omitting the comma).

- If two vectors are requested the second vector must start on a new line .

- An empty string (obtained by just pushing the RETURN button) is considered to mean zero whenever a single number (Integer or Real) is requested. This is particularly useful when a zero entry implies that a submode (see the descriptions of the PLOT and the USER commands in sections 4.14 and 4.15 respectively) is to be exited.

- When two integers are requested either one or both can be represented by an empty string. (The only command for which this is useful is TYPE.)

- Floating Point numbers (like 1.0E-12 or 1D-4) are processed correctly.

- Floating Point numbers may also be represented as integers.

After receiving the data, MICROSCOPE processes the command and the data. If the screen is active the command is carried out and the alphanumerical display is updated, otherwise the next command is prompted for.

If a command is not recognized a message is printed to that effect and a new command is prompted for. If data are requested, and a number is not recognized, then new data will be requested. However, if data are syntactically correct, but otherwise meaningless (e.g. you request the graph of the 117th derivative), then the command will be aborted and a new command will be prompted for.

There are currently 63 commands which divide into 11 major groups corresponding to sections 4.5 through 4.15. Rather than describing the groups in this section, we illustrate some of them in the course of a search for a point of reduced smoothness (the 'active point'), giving the names of the commands used, and further illustrating the capabilities of MICROSCOPE.



### 3.3 A Discontinuity in the second derivative

The trig function underlying this example is

$$f(x) = (\sigma^2 x^2)/2 + c \quad \text{if } x > 0 \text{ and } f(x) = c \quad \text{otherwise}$$

(where  $\epsilon = -0.005$ ). Clearly  $f$  is once but not twice differentiable at the origin. The exponential term serves to conceal the discontinuity in the second derivative. (Similar functions that are 0 through 5 times differentiable can also be examined for arbitrary values of  $\epsilon$ .) The parameters are changed by giving the user intervention command `USER` (assuming the test package has been loaded, see appendix III and section 5).

Let us pretend we do not know the the degree of smoothness but we wish to examine it. We first inform **MICROSCOPE** that the trial function has one independent variable, using the **DMNSN** command. Then we enter a search interval (here  $[-1,3]$ , say) using the **IINTVL** command. At this stage, the screen is still inactive. This is changed, and the first display is obtained, by giving the **GO** command. The display will remain active throughout the remainder of this section. We obtain:

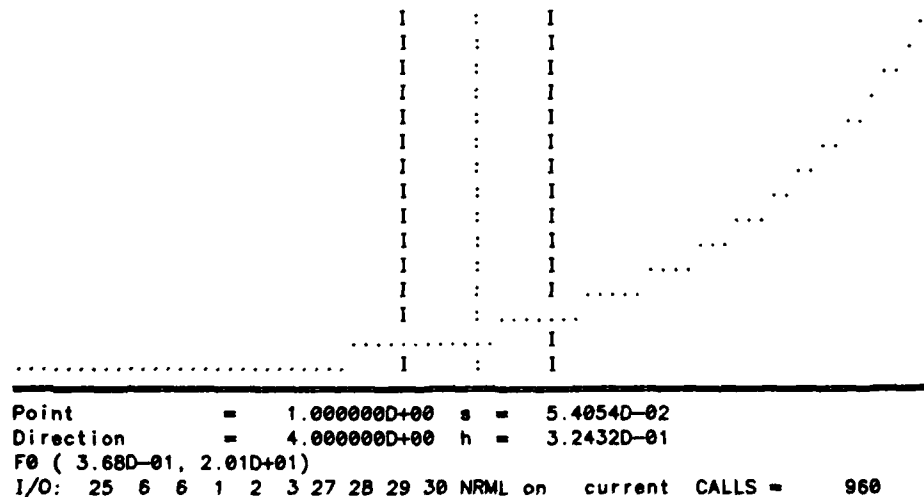


Figure 13

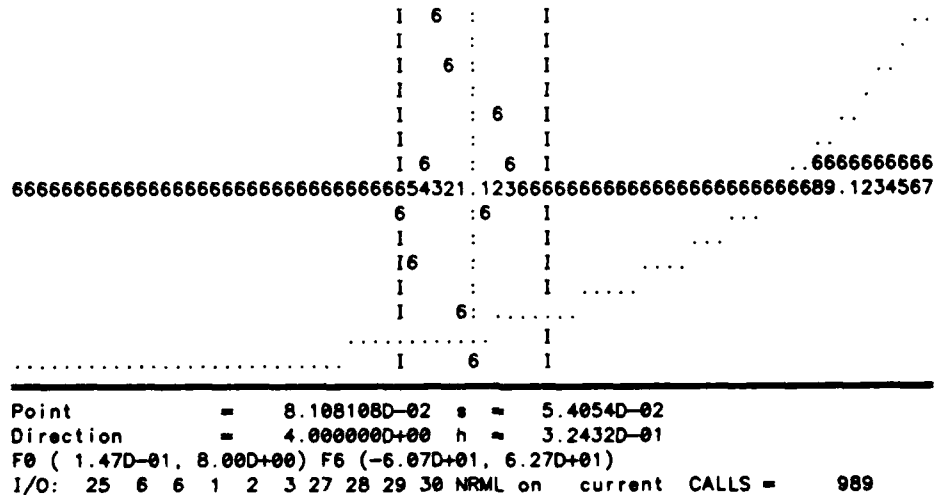
This display shows no lack of smoothness at all. So, not knowing about the nature of the suspected discontinuity at all we turn on the horizontal scale (using DSCALE) to locate any phenomena of interest and activate (using DGRAPH) the graph of the highest possible (i.e. sixth) derivative. These changes are incorporated one by one (not shown here) into an active screen. The final result is this:

[illegible]

```
Point      = 1.000000D+00 s = 5.4054D-02
Direction = 4.000000D+00 h = 3.2432D-01
F0 ( 3.68D-01, 2.01D+01) F6 (-6.07D+01, 6.27D+01)
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 972
```

Figure 14

Now it is clear that there is a phenomenon of some kind occurring at the point -17 measured on the horizontal scale. To examine this further, we shift the graph by 17 (using SHIFT). This centers the interesting parts of the graph, yielding:



**Figure 15**

Already at this stage it is possible to tell the nature of the reduced smoothness. The sixth derivative clearly shows four distinct local extrema. This qualifies it as the third (numerical) derivative of a delta function. The second derivative of the trial function must therefore be a step function. The trial function is hence once but not twice differentiable. Derivatives of step functions are discussed in detail in section 4.2 and in appendix I.

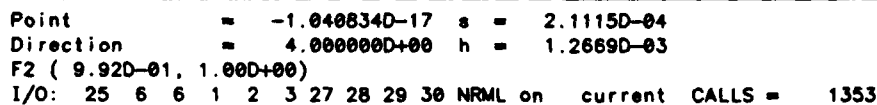
To confirm our diagnosis directly we examine the second derivative. Turning on its graph and erasing the distracting graph of the trial function (using EGRAPH), however, yields the following display which shows no lack of smoothness at all in the second derivative:

[illegible]

```
Point      = 8.108108D-02  s = 5.4054D-02
Direction  = 4.000000D+00  h = 3.2432D-01
F2 ( 1.48D-01, 8.08D+00) F6 (-6.07D+01, 6.27D+01)
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 989
```

**Figure 16**

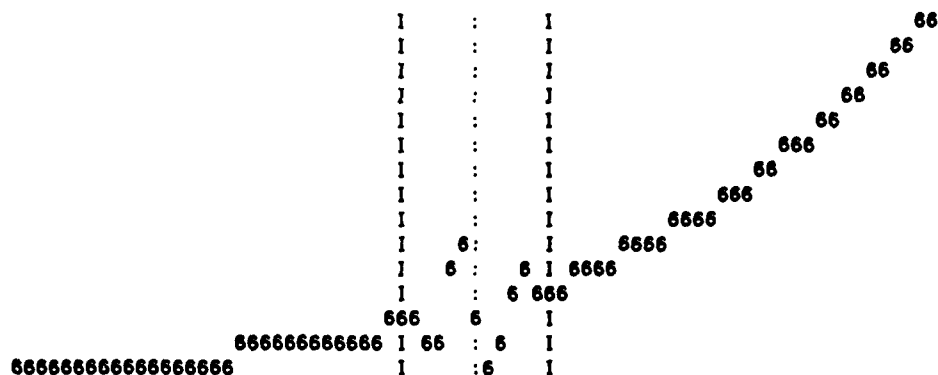
The second derivative appears smooth because the discretization is too crude to show its discontinuity. So we keep decreasing the value of the discretization parameter (and simultaneously the interval, keeping the window width fixed). Since the active point is only approximately in the center of the display, this will cause it to drift to one side which has to be counteracted by more shift commands. Because of the organization of MICROSCOPE it is particularly efficient and convenient to divide (or multiply) with powers of two. Halving  $h$  four times (using HALVE) followed by a shift of 24 and four more halvings yields, after erasing the scale and the graph of the sixth derivative:



Now it is clear that there is a jump discontinuity in the second derivative of the trial function. The location of the active point can be seen to be  $\theta \pm$  about  $s = 0.0002$ . Even the size of the jump can be estimated. Inquiring (using TYPE) for the values of the derivative at the left and right end points of the window (where in this crude display it assumes its extrema) yields values of  $9.98846D-1$  and  $9.96522D-1$  respectively, corresponding to a jump of  $-2.32D-3$ . The real jump is of course  $\eta = -5.0D-3$ . The discrepancy is due to a truncation error, not just in approximating the derivative, but also in picking the points of evaluation. The estimate can be improved by extrapolating (linearly) the left and right parts of the second derivative to the point of examination, using an estimate of the third derivative which can be obtained from one of the earlier displays. With the third derivative set equal to 1, this procedure yields an estimate of  $-2.32D-3 - 12s = -4.86D-3$  which is a very good result considering the difficulties of numerical differentiation.

In the present setting, it is not possible to decrease the discretization parameter further without succumbing to round-off errors in the second derivative (you may wish to try this yourself). However, the above example illustrates how to alleviate the limitations of round-off errors. We already recognized the true nature of the reduced smoothness when examining the sixth rather than the second derivative. With the accuracy used for the examples in this manual (10 digits), steps in the second derivative 300 times as small as the one above can be detected. The following display shows the graph of the sixth derivative for the same

function as above with  $\eta = -1.70-5$  and  $h = 0.24$ . These values have been taken from table 4 in appendix I ( $\tau = 1.00-10$ ).



Point = -1.040834D-17 s = 4.0000D-02  
 Direction = 4.000000D+00 h = 2.4000D-01  
 F6 ( 2.23D-01, 4.40D+00)  
 I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 1440

Figure 18 At the limits of resolution

The sixth derivative shows 4 extrema, identifying the step in the second derivative.

#### 4. A Detailed Description of Features and Commands

##### 4.1 Round-Off Effects

The fact that all arithmetic operations are carried out with finite accuracy is fundamental and all pervasive in working with MICROSCOPE. Numerical Differentiation is particularly sensitive to rounding since it involves the subtraction of nearly equal quantities, leading to a cancellation of significant digits. Consequentially, the error in numerical differentiation does not decrease indefinitely with the discretization parameter, but rather starts to behave erratically beyond a certain minimum value and tends to increase as the discretization parameter is decreased further. However, it is its very randomness that makes round-off errors readily apparent in a MICROSCOPE display so that you will not be misled into accepting meaningless results at their face value.

The precise effect of rounded arithmetic depends on several factors:

-1- The number of significant (binary) digits carried by a particular computer. Most users will prefer an approximation of the equivalent number of decimal digits, and so MICROSCOPE computes that number and prints it out at the beginning of a session. More precisely, MICROSCOPE computes the smallest power of 2, tau, say, such that within the given computer arithmetic the quantity  $1 + \tau$  satisfies a test for greater than 1, and then prints the rounded value of the negative common logarithm of tau.

-2- The accuracy with which the trial function is being evaluated. That accuracy will always be less than or at most equal to the accuracy indicated in -1-. Usually it will be much less, for any of the following reasons:

-2.1- Typical multivariate interpolants are very complicated algebraically, involving many operations which tend to increase round-off errors.

-2.2- If, as is frequently the case, the trial function is the difference between a primitive function and its interpolant, then a cancellation of significant digits occurs.

-2.3- If the display function is a cross derivative of the trial function it is already contaminated by round-off errors due to numerical differentiation, thus exhibiting effects that one would normally expect to occur only for higher order tangential derivatives.

-3- The precise nature of round-off effects is also very sensitive to the relative magnitudes of various tangential derivatives involved. For example, a constant function will exhibit round-off errors at any value of the discretization parameter.

Of the above three sources of round-off errors, you will usually have a firm knowledge only of the first. Knowledge about the other two may be gained by using MICROSCOPE itself. This is usually more effective than attempting a theoretical analysis.

The precise nature of round-off errors is discussed in Appendix I. Detailed formulas and tables are given. However, the usual approach will be to experiment with various values of the discretization parameter, choosing it as small as possible without generating the telltale random pattern that indicates round-off errors.

#### 4.2 Derivatives of Step Functions

One perhaps surprising feature of numerical differentiation is that doubtful discontinuities in some derivatives can often be displayed very clearly by plotting higher order derivatives, even though a large discretization parameter is necessary to keep them free of round-off errors. This fact becomes quickly apparent empirically, and is analyzed in Appendix I.

The conceptual basis for the approach is provided by a Dirac Delta function (see Courant and Hilbert, 1962, p.456). A step function can be thought of as rising at the discontinuity with infinite slope from one discrete value to another. If it is piecewise constant its derivative can be thought of as a function that is zero everywhere, except at the discontinuity, where it is infinite such that the integral over the derivative equals the jump in the step function. That derivative is a delta function.

More precisely, a delta function about a point  $P$ , say, is defined as the limit of a sequence of non-negative functions that are different from zero only in neighborhoods of  $P$ . The diameter of those neighborhoods tends to zero, and the value of the integral over the functions remains constant (at 1 in the classical definition).

Derivatives of a delta function can be visualized by considering a function in the above sequence. Suppose it is bell shaped. Then its derivative will have two local extrema of opposite signs. The second derivative will have three extrema, etc. Taken to the limit, the  $k$ -th derivative of a step function has  $k$  local extrema of alternating signs, all located at the same point.



Numerical derivatives of a step function exhibit the same phenomena, if the discontinuity is smeared out over a window containing sufficiently many points. This is illustrated in the pictures below. The step discontinuities occur in the  $m$ -th derivatives, where  $m = 0, 1, \dots, 5$ . The discretization parameter is  $h = 1$ , and the size of the jump is 1. More precisely, the  $m$ -th function is defined by

$$s(x) = 0 \text{ if } x < 0 \text{ and } x/m! \text{ otherwise,}$$

where the exclamation mark denotes the factorial.

Notice that as the degree of the discontinuous derivative increases, the smoothness of the graphs of higher derivatives increases. This is consistent with the interpretation of numerical differentiation as a smoothing process. (The numerical derivatives have the same degree of smoothness as that of the display function, just the number of the points of reduced smoothness is increased.)

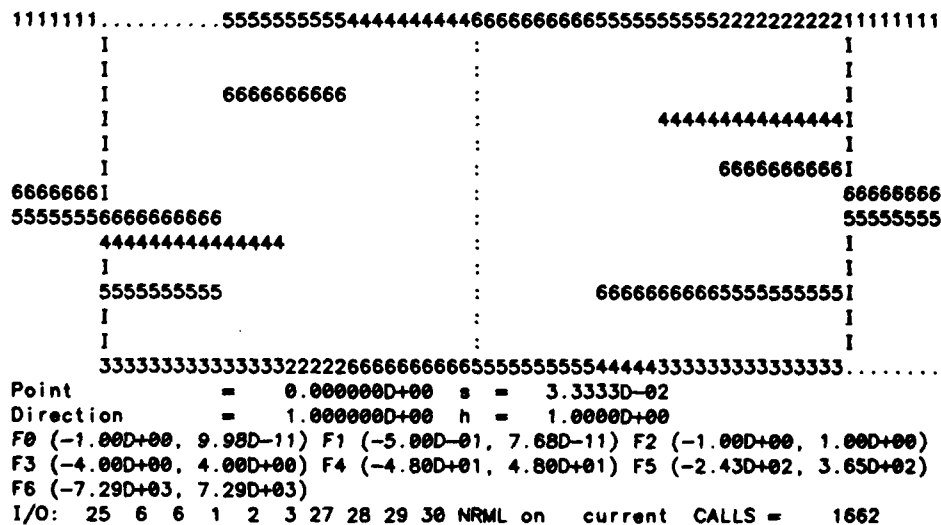


Figure 19 Derivatives of a Step Function ( $m = 0$ )

```

222222222211      665      444      86 33333      2222222222
I 221111      65 65      4 : 4      6 36      33      22 I
I 22 1111      665 655 4 : 4      633 86      33 22 I
I 22 11615      54 : 4 33      6 5 33 I
I 22 6 51111 6 45 : 436      6 5255533 I
666666666      266 5      1116 55 : 33 6      662 5553 66666666
1666666      6 25      4 11115:33 4      22 6 666666I
555555555      6666 5 22      4 6 3511 6 4      22 5 6666 55555555
I 555      5 22 4      633 :551611 4 22 5 I
44444444      355      24 33 : 5 11142 5      44444444
I444 3555 5      4 2233 6 : 65 22 4115      444I
I 444 335      4 3322 6 : 6 552 45 1111      444 I
I 44433      4 33 22 : 22 5 54 11444 I
I 4443 433      226:622 5 5 4 444 1111 I
I 4443      262 55 444 111111111

```

Point = 0.000000D+00 s = 3.3333D-02  
Direction = 1.000000D+00 h = 1.0000D+00  
F1 (-1.00D+00, 4.28D-11) F2 (-1.00D+00, 2.85D-10) F3 (-2.00D+00, 2.00D+00)  
F4 (-8.00D+00, 1.60D+01) F5 (-8.10D+01, 8.10D+01) F6 (-1.46D+03, 9.72D+02)  
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 1797

0  
Figure 20 Derivatives of a C Function (m = 1)

```

3333333333332222      66666555555      4444444      3333333333333
I 33 22222      6 56 : 544      44      33 I
I 33 2222      5 : 45      4 33 I
I 3 62 5 6 : 4 5      44 3 I
I 33 6 252 : 4 5      644 I
I 3 5 22 6: 5      666 6664 I
I 3 6 222:4      6 3 6666 I
6666666666666      3 6 5      62 5 6 3      6666666666666
555555555      4666      6 5      4: 22 5 6 3      555555555
I 555 46666666 3 :6 22 3      555 I
I 55 4 35      4 : 222 63      55 I
I 55 44 5 3      4 : 6 2625      55 I
I 5 4 33 4 : 33 222 5 I
I 5 455 344 : 6 33 6 55222225 I
I 55555 4444444 3333333336666      5555522222222222222222

```

Point = 0.000000D+00 s = 3.3333D-02  
Direction = 1.000000D+00 h = 1.0000D+00  
F2 (-1.00D+00, 1.99D-10) F3 (-1.00D+00, 1.74D-09) F4 (-2.67D+00, 2.67D+00)  
F5 (-8.98D+00, 1.35D+01) F6 (-1.46D+02, 1.46D+02)  
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 1932

1  
Figure 21 Derivatives of a C Function (m = 2)

[illegible]

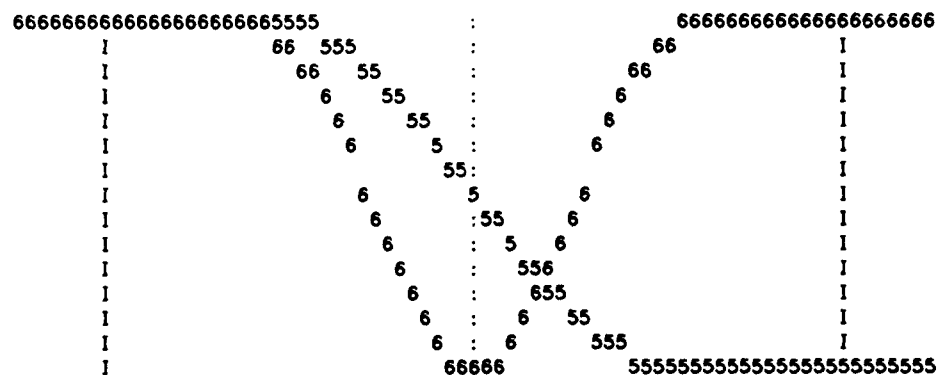
```
Point          = 0.000000D+00 s = 3.3333D-02
Direction      = 1.000000D+00 h = 1.0000D+00
F3 (-1.00D+00, 1.60D-09) F4 (-1.33D+00, 1.14D-08) F5 (-3.00D+00, 3.00D+00)
F6 (-1.28D+01, 2.70D+01)
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 2067
```

Figure 22 Derivatives of a  $C^2$  Function ( $m = 3$ )

[illegible]

```
Point      = 0.000000D+00  s = 3.3333D-02
Direction  = 1.000000D+00  h = 1.0000D+00
F4 (-1.00D+00, 9.60D-09) F5 (-1.38D+00, 6.61D-08) F6 (-4.13D+00, 4.13D+00)
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 2202
```

Figure 23 Derivatives of a  $C^3$  Function ( $m = 4$ )



```
Point          = 0.000000D+00 s = 3.3333D-02
Direction      = 1.000000D+00 h = 1.0000D+00
F5 (-1.00D+00, 5.09D-08) F6 (-1.65D+00, 2.67D-06)
I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 2337
```

Figure 24 Derivatives of a C<sup>4</sup> Function (m = 5)

### 4.3 The Window

Sometimes it is desirable to use very small window widths or even completely separate differentiation stencils, for example when searching a large domain for a point of reduced smoothness. Then steps in a derivative will appear as such rather than as gradual slopes that might be obscured by other features of the relevant graphs. Also, cusps and poles are best displayed using small windows (see appendix III for some examples). Usually, however, it will be preferable to use larger windows.

There are two main reasons to use overlapping differentiation stencils: first, as illustrated in the preceding subsection, this is necessary to resolve all the extrema of a derivative of a delta function, and, second, it allows the use of one evaluation of a trial function for several values of a derivative, thereby increasing efficiency and accelerating your interaction with MICROSCOPE. Using overlapping stencils is equivalent to having a window width ( $w = 2h/s$ ) that is greater than 1.

Due to the internal organization of **MICROSCOPE**, for any given window width, the number of evaluations (of the display function) required is determined solely by the largest degree of the plotted derivatives. The following table lists formulas for computing the number N of evaluations

required given the window width ( $w$ ), the highest degree of a plotted derivative ( $k$ ) and the number of columns ( $C$ ) in the graphical display.

max k:	0	1 or 2	3 or 4	5 or 6
$w=2h/s$				
< 1	C	3C	5C	9C
2	C	C + 2	2C + 3	4C + 5
4	C	C + 4	C + 4	3C + 8
6	C	C + 6	2C + 9	2C + 9
8	C	C + 8	C + 9	3C + 16
10	C	C + 10	2C + 15	4C + 25
12	C	C + 12	C + 12	C + 12
14	C	C + 14	2C + 21	4C + 35
16	C	C + 16	C + 16	3C + 32
18	C	C + 18	2C + 27	2C + 27
20	C	C + 20	C + 20	3C + 40
22	C	C + 22	2C + 33	4C + 55
24	C	C + 24	C + 24	C + 24

Table: Efficiency of Window Widths

The special role played by  $w = 12$  is obvious: the numerical effort is independent of the degree of the plotted derivatives, and exceeds the number of columns in the display only by a constant. This is true whenever  $w$  is a multiple of 12. The reason for this is, of course, that the display function is sampled at multiples of  $1/2$  and  $1/3$  of the discretization parameter. For efficiency, the default value of  $w$  is 12. This is also sufficient for most examinations involving delta functions. Sometimes, however, a larger multiple, like 24 or 36 may be desirable.

#### 4.4 Reference Table of Commands

This subsection contains a quick reference table of all available commands. For each command, a one-line description is given. Often, the information contained in these short descriptions will be all that is needed. Therefore, the descriptions are used in several other places. An appropriate selection of them occurs at the beginning of each of the following subsections. The HSUMRY command lists them ordered by topics. The more detailed HELP command gives a documentation of an individual command headed by the one-line description.

Name	Description	Data	Error	Screen
ACCENT	mark derivative with asterisks TOGGLE	k	-1<k<7	immediate
C1CROSS	read 1st component of cross direction	C1	C>0	deactiv.
C2CROSS	read 2nd component of cross direction	C2	C>0;1	deactiv.
C3CROSS	read 3rd component of cross direction	C3	C>0;1	deactiv.
CCHANNEL	change channel number	ic,nc	4	—
CDIRCTN	read direction of cross direction	C	C>0	deactiv.
CHVALUE	read ch for cross differentiation	ch	5	deactiv.
CORDER	read order of cross derivative	k	-1<k<7	deactiv.
CWINDOW	change window width by factor SIGN	m	m=0	immediate
D1CHDIR	read 1st component of dir. of inv.	D1	D>0	deactiv.
D2CHDIR	read 2nd component of dir. of inv.	D2	D>0;1	deactiv.
D3CHDIR	read 3rd component of dir. of inv.	D3	D>0;1	deactiv.
DCENTER	draw center of graph. display TOGGLE	—	—	immediate
DGRAPH	draw graph	k	-1<k<7	immediate
DIVIDE	divide h by integer m SIGN	m	d>0	immediate
DMNSN	read dimension v of domain	v	0<v<4	deactiv.
DOUBLE	double h	—	—	immediate
DSCALE	draw scale in graphical display TOGGLE	—	—	immediate
DXAXIS	draw horizontal axis TOGGLE	—	—	immediate
EGRAPH	erase graph	k	-1<k<7	immediate
EXIT	return to calling program	—	—	deactiv.
FLIP	replace d by -d TOGGLE	—	—	immediate
FORCE	recompute everything	—	—	activ
GO	compute whatever is needed	—	—	activ
HALVE	halve h	—	—	immediate
HELP	obtain detailed help on command cmd	cmd	2	—
HSUMRY	obtain a summary of all commands	—	—	—
IDIRCTN	read the direction of investigation	D	D>0	deactiv.
IHVALUE	read value of h	h	h>0	deactiv.
IINTVL	read endpoints of line of investig.	A,B	A>B	deactiv.
IPOINT	read the new point of examination	P	—	deactiv.
LCROSS	load a new cross direction	kc	3	deactiv
LDRCTN	load a new direction of investigation	kd	3	deactiv
LIST	print all currently available commands	—	—	—
LOG	give channel number for logging	nc	—	—
LPOINT	load a new point of examination	kp	3	deactiv.

MULTPLY	multiply h by factor m	SIGN	m	m>=0	immediate
NEWS	print the news	---	---	---	---
NORMAL	turn normalization of D on/off	TOGGLE	---	---	deactiv.
OUTPUT	write screen image on recording device	---	---	---	---
P1CHPNT	read first component of P	P1	---	---	deactiv.
P2CHPNT	read second component of P	P2	1	---	deactiv.
P3CHPNT	read third component of P	P3	1	---	deactiv.
PAUSE	pause until RETURN from channel nc	nc	---	---	---
PLOT	enter plotting (<PLOT79>) mode	---	---	---	---
QUIT	terminate MICROSCOPE session	---	---	---	---
RCROSS	read channel number for cross list	nc	---	---	---
RDIRCTN	read channel number for direction list	nc	---	---	---
RESTART	reestablish earlier parameter settings	---	---	---	---
ROTATE	change direction of investigation	DD	D>=0	---	deactiv.
RPOINT	read channel number for point list	nc	1	---	---
RSCREEN	refresh screen	---	---	---	pending
RWIND	rewind device nc	nc	---	---	---
SETDF	reset defaults	---	---	---	deactiv.
SHIFT	shift point of examination	is	---	---	immediate
STORE	store current parameter settings	---	---	---	---
TCENTER	type value of k-th derivative at center	k	-1<k<7	---	---
TNOTE	take a note on recording device	---	---	---	---
TYPE	Type value of derivative at some point	k,is	-1<k<7	---	immediate
UNDO	undo previous parameter changes	---	---	---	---
USER	enter user mode	---	---	---	---
WAIT	make program wait for GO or FORCE com.	---	---	---	deactiv.
ZOOM	change window width by factor	SIGN	m	m>=0	immediate

Quick Reference Table of commands

Key to table of commands:

— Description —

TOGGLE	the command reverses itself if called again
SIGN	means that reversing the sign of the input parameter generates the opposite effect (e.g. multiply instead of divide).

— Data —

A	one endpoint of the line of investigation
B	the other endpoint of the line of investigation ( $P = (A + B)/2$ )
C	cross direction
C1	first component of cross direction
C2	second component of cross direction

C3	third component of cross direction
ch	discretization parameter for cross derivatives
cmd	command name
D	direction of investigation
D1	first component of direction of investigation
D2	second component of direction of investigation
D3	third component of direction of investigation
DD	correction to be added to D
ic	type of device (1 = input, 2 = output, 3 = graphic, 4 = record, 5 = restart)
is	integer shift in number of columns on display, +ve shifts graph to right, point to left, -ve is reverse
k	degree of a derivative (to be plotted, erased, etc.)
kc	position of the cross direction in a list
kd	position of the direction of investigation in a list
kp	position of the point of examination in a list
m	integer multiplier.
n	integer divider
nc	channel (device) number
P	point of examination
P1	first component of the point of examination
P2	second component of the point of examination
P3	third component of the point of examination
—	no data are required

— Error —

Restrictions on the data are listed in this column. The entry "—" means that no check is made for errors. The symbol "><" means "not equal". If the error conditions cannot be given explicitly then the following reference numbers have been used:

1. Components to be changed must not have an index that is greater than the current dimension.
2. The command must be a valid MICROSCOPE command. If it is not recognized you will be prompted for another command to be helped on. Typing any valid command will get you some information on it and will then return you to command mode.
3. There must be one vector in legal format in each line of the list. If illegal characters are encountered, an error termination of the command will occur. Also, the index number must not exceed the number of lines in the list. If it does, the program will reach the end of the file containing the list, and the precise action after that depends upon the computing system.

— Screen —



activ            the command activates the screen

deactiv.        the command deactivates the screen - i.e. the GO or FORCE command must be given before the computation proceeds. For some commands, there is a check if the newly input datum is identical with the old one, in which case the screen remains active.

immediate      If the screen is active then the effects of the command are incorporated before the next command is requested

pending        If the screen is inactive, this command will put the graphical display on the screen, even if computations are pending.

—              The graphical and numerical display, and the screen status are not changed. Notice, however, that the actual screen display may be destroyed by error messages or help information. The next display changing command (or RSCREEN) will then refresh the screen.

#### 4.5 Commands to Control the Display

In this and the following subsections, input to MICROSCOPE is via the assigned input channel, unless otherwise stated. This will usually be the terminal, but may also be a file for a batch type session.

ACCENT	mark derivative with asterisks	TOGGLE	k	-1<k<7	immediate
DCENTER	draw center of graph. display	TOGGLE	—	—	immediate
DGRAPH	draw graph		k	-1<k<7	immediate
DMNSN	read dimension v of domain		v	0<v<4	deactiv.
DSCALE	draw scale in graphical display	TOGGLE	—	—	immediate
DXAXIS	draw horizontal axis	TOGGLE	—	—	immediate
EGRAPH	erase graph		k	-1<k<7	immediate
FLIP	replace d by -d	TOGGLE	—	—	immediate
RSCREEN	refresh screen		—	—	pending
TCENTER	type value of k-th derivative at center		k	-1<k<7	—
TYPE	type value of derivative at some point		k, is	-1<k<7	immediate

We now turn to the descriptions of individual commands. Screen display commands control what appears on the terminal screen. (For the effects of these commands on the <PLOT79> drawing see section 4.14.) There are several subgroups of commands:

-1- Deciding on the Tangential Derivatives to be Displayed.

By default, only the display function is displayed. The graph of the display function itself is marked by periods, the graph of its  $k$ -th tangential derivative ( $k = 1, 2, \dots, 6$ ) by the digit  $k$ . Using DGRAPH or EGRAPH respectively, you can turn on or off the graphs of the zeroth through sixth derivatives. It is also possible to display no functions at all. In any case, the display function is evaluated only at the points where it is needed.

#### -2- Accentuating a Graph.

If you wish to emphasize the graph of a particular derivative then you can plot its graph using asterisks rather than a period or the digit  $k$ . The command ACCENT turns this option on or off. Note that the numerical display contains no direct information on which derivatives have been accentuated. Also, if you accentuate several derivatives at once their graphs cannot be distinguished from each other. The graph of an accented derivative overwrites anything else in the display, except for the center mark. The ACCENT command also has the effect of turning the graph of a derivative on, i.e. it does not need to be preceded by a DGRAPH command. By default no derivative is accented.

#### -3- Drawing a Horizontal Axis

A horizontal axis can be drawn through the center of the graphical display, using the command DXAXIS. If desired, a scale can be obtained by using DSCALE. Both commands are on/off switches. The point at which the axis or scale intersects the central column of the graphical display does not in general correspond to the value zero. (That point is different for each of the tangential derivatives plotted, and may not even be contained in the graphical display.) The purpose of the axis is to give help in examining symmetry properties, and the scale is useful in locating points of interest for more detailed examination. Both the scale and the axis are off by default.

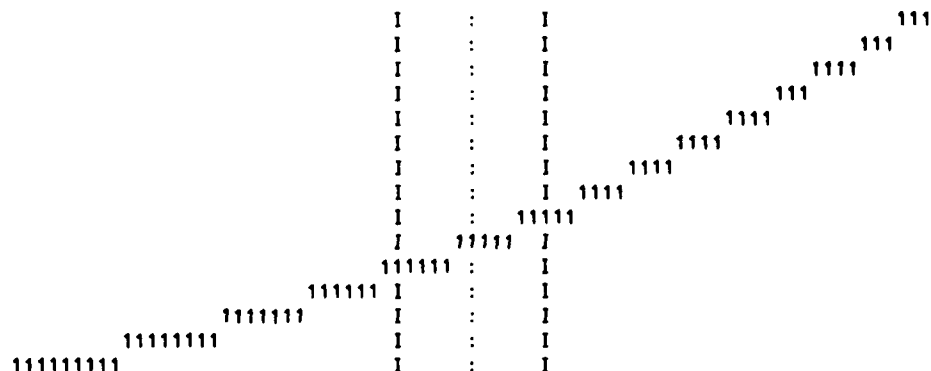
#### -4- Marking the Center

The on/off command DCENTER may be used to mark the center of the graphical display with a plus sign. This is sometimes useful for confirming symmetry properties. The center mark is off by default.

#### -5- Flipping the Direction of Investigation

The FLIP command has the effect of replacing the direction of investigation  $d$  by  $-d$ . This turns the graph of any derivative about the vertical axis, and it also turns the graph of any odd degree derivative about the horizontal axis, thereby possibly disentangling graphs that have been overwritten by others.

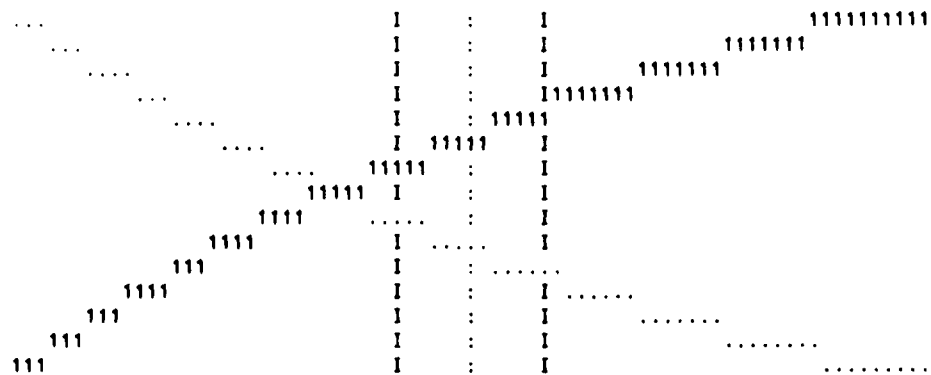
Consider for example the exponential function and its first derivative. Ordinarily the two graphs would be identical, yielding a display in which the display function is completely covered by the graph of its first derivative:



Point = 0.000000D+00 s = 1.6667D-02  
 Direction = 1.000000D+00 h = 1.0000D-01  
 F0 ( 5.40D-01, 1.85D+00) F1 ( 5.41D-01, 1.86D+00)  
 I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 2424

Figure 25 An Invisible Graph

After using the FLIP command one obtains:



Point = 0.000000D+00 s = 1.6667D-02  
 Direction = -1.000000D+00 h = 1.0000D-01  
 F0 ( 5.40D-01, 1.85D+00) F1 (-1.86D+00,-5.41D-01)  
 I/O: 25 6 6 1 2 3 27 28 29 30 NRML on current CALLS = 2424

Figure 26 The Disentangled Graphs

where the two graphs are now clearly visible.

#### -6- Obtaining Numerical Values

The alphanumerical display does not provide precise numerical values. Values of any computed derivative to 21 digits may be obtained by using the TCENTER or the TYPE command. TCENTER provides values at the center of the display, TYPE does so for any point in the display. For TYPE, measure the distance in columns from the center of the graphical display, positive to the right and negative to the left. For clarity, if the screen version of MICROSCOPE is employed, the column in which the value is read is briefly marked with exclamation marks. Returning to command mode requires typing a carriage return. This is necessary so that the prompt for the next command does not overwrite the numerical information before it can be read.

Only derivative values that have been previously computed can be read. This includes all derivatives through [k] where k is the highest degree of a derivative that has been previously displayed on the current active screen, and [k] is k+1 if k is odd or k if k is even.

#### -7- Refreshing the screen

The command RSCREEN has two distinct applications:

On an active screen, it serves to refresh the screen image after it has been destroyed e.g. by a system message or by HELP information.

On an inactive screen, the last few commands instead of the alphanumerical display are shown. In that situation, RSCREEN calls the display onto the screen without carrying out any computations. Thus changed parameters may be shown in the numerical display without having been incorporated into the graphical display. This condition is indicated by the flag "GO pndg" in the numerical display. (If the screen is active then the flag "current" is displayed instead). The second application is useful if the alphanumerical display has to be viewed in order to decide on the next command.

When entering MCRSCP the screen is inactive.

#### -8- Setting the Domain Dimension

The DMNSN command informs MICROSCOPE of the number of independent variables (1,2 or 3). It is listed here because it does affect the numerical display. However, this command is usually given at most once at the beginning of the MICROSCOPE session. The default dimension is 2. DMNSN also resets the point of examination, the direction of investigation, the cross direction, and the order of the cross derivative to their default values.

#### 4.6 Commands to Control the Point of Examination

IINTVL	read endpoints of line of investig.	A,B	A<B	deactiv.
IPOINT	read the new point of examination	P	—	deactiv.
LPOINT	load a new point of examination	kp	3	deactiv.
P1CHPNT	read first component of P	P1	—	deactiv.
P2CHPNT	read second component of P	P2	1	deactiv.
P3CHPNT	read third component of P	P3	1	deactiv.
RPOINT	read channel number for point list	nc	—	—
SHIFT	shift point of examination	is	—	immediate

The simplest way of defining the point of examination is to read it from the input device by using the IPOINT command. Individual components may be defined similarly by using the commands P1CHPNT, P2CHPNT or P3CHPNT.

Sometimes it is useful to generate a list of possible points of examination before the MICROSCOPE session, and then refer to them by their position (i.e. line number) in the list, rather than by typing the

individual components. This is facilitated by the LPOINT command, which prompts for the position of the point and then reads it from a device (usually a file) with a device number that has been set by the RPOINT command (the default is zero). All point loading devices must contain one vector per line.

The SHIFT command can be used to shift the graph by a specified number of columns to the right (if the number is positive) or left (if it is negative). There is no restriction on the number of columns by which to shift the graphs. In particular, the number may exceed the number of columns in the display.

It is also possible to define the part of the domain that is to be shown in the display by defining its two endpoints. This is accomplished by the IINTVL command, which will prompt for the left and right endpoint (in that order) of the graphical display. These have to be separated by at least one line feed. Notice that IINTVL defines simultaneously the point of examination, the direction of investigation, the discretization parameter, and the display interval.

#### 4.7 Commands to Control the Direction of Investigation

D1CHDIR	read 1st component of dir. of inv.	D1	D>0	deactiv.
D2CHDIR	read 2nd component of dir. of inv.	D2	D>0;1	deactiv.
D3CHDIR	read 3rd component of dir. of inv.	D3	D>0;1	deactiv.
IDIRCTN	read the direction of investigation	D	D>0	deactiv.
LDRCTN	load a new direction of investigation	kd	3	deactiv.
NORMAL	turn normalization of D on/off TOGGLE	---	---	deactiv.
RDIRCTN	read channel number for direction list	nc	---	---
ROTATE	change direction of investigation	DD	D>0	deactiv.

The direction of investigation can be read from the input device by using the command IDIRCTN. Individual components can be changed by the commands D1CHDIR, D2CHDIR or D3CHDIR.

Similarly to the point of examination, the direction of investigation may also be defined by giving its position in a list. The command LDRCTN loads the direction from the list (usually a file) whose channel number has been defined by the RDIRCTN command.

A new direction may also be defined by rotating the old one. The command ROTATE will read the difference between the old and the new direction, and update the direction. This is sometimes useful for examining the effects of deviating slightly from the previous direction of investigation.

The numerical display will always contain the direction of investigation as it has been input to MICROSCOPE. Internally the direction of the derivative (i.e. the vector  $d$  in the differentiation formulas) is

usually the given direction divided by its Euclidean length. This leads to the standard definition of a directional derivative. However, sometimes it is more useful to use the unnormalized direction for the differentiation (i.e. to have  $d$  identical to the displayed numerical value). The normalization can be turned on or off using the command `NORMAL`, and is on by default. Normalization changes the values of the directional derivatives, and the domain that is covered by the graphical display. An identical graphical display (but with different numerical values) can be obtained by dividing the display interval instead of the specified direction of investigation by the same factor.

#### 4.8 Commands to Control Accuracy, the Window, and Round-Off Effects

CWINDOW	change window width by factor	SIGN	m	m=0	immediate
DIVIDE	divide h by integer m	SIGN	m	d>0	immediate
DOUBLE	double h	---	---	---	immediate
HALVE	halve h	---	---	---	immediate
IHVALUE	read value of h		h	h>0	deactiv.
MULTPLY	multiply h by factor m	SIGN	m	m>0	immediate
ZOOM	enlarge central portion of displ.	SIGN	m	m>0	immediate

This group of commands controls the three parameters  $h$  (discretization parameter),  $s$  (the display interval), and  $w$  (the window width) which are related by  $w = 2h/s$  ( $w$  must lie in the interval  $[1/16, 576]$  and can assume only certain rational values). Each of the commands changes two of the three parameters, leaving the third constant, according to the following table:

Parameter left constant		
w	s	h
DIVIDE	CWINDOW	ZOOM
DOUBLE		
HALVE		
IHVALUE		
MULTPLY		

Table of Discretization Controlling Commands

Most frequently, the window width is left unchanged, often throughout the entire `MICROSCOPE` session. This accounts for the abundance of commands covering that case.

The command `IHVALUE` serves to define an entirely new value of the discretization parameter (and the display interval).

Often you will want to modify the current value of  $h$ , e.g. to get rid of the influence of round-off errors, or perhaps to increase the accuracy of numerical values. Due to the organization of `MICROSCOPE` it is particularly efficient (i.e. requires few evaluations of the trial

function) if  $h$  is multiplied or divided by powers of 2. The factor 2 is used so frequently that two specific commands (HALVE and DOUBLE) have been provided for this case, which require no arguments.

Multiplications or divisions with other integer factors can be accomplished by using the commands MULTIPLY and DIVIDE respectively. It is possible to reverse the meaning of the commands by entering negative factors (e.g. MULTIPLY followed by -3 is equivalent to DIVIDE followed by +3). This is due more to the internal workings of MICROSCOPE than to any attempt to provide a truly meaningful convenience.

The command CWINDOW changes the current window width by the specified factor. A positive factor corresponds to an increase of the window width, a negative one to a decrease. The change is obtained by altering the value of the discretization parameter while leaving the display interval fixed.

The command ZOOM has a similar effect. However, it alters the display interval and leaves the discretization parameter unchanged. This is useful when operating close to the round-off level with a value of  $h$  that must not be decreased. On the other hand, if round-off errors are irrelevant then the CWINDOW command should be used because it is more efficient.

The primary purpose of the commands described here (other than IHVALUE is to alter the display while the screen is active. If several commands are given while the screen is inactive only the last one will actually be implemented. If several changes are to be effected before the trial function is evaluated, then this can be accomplished by erasing the graphs of all derivatives, and then activating the screen (using GO or FORCE), and making the desired changes.

For the following illustration it is assumed that the screen is active derivatives through second order have just been displayed, 79 columns are being carried in the graphical display, and the current window width is 12. The table below gives the additional number of evaluations of the display function needed to accommodate changes by factors 2,3,...,8. An entry "-" indicates that the change is impossible (e.g. repeated applications of the HALVE command can only accomplish changes by powers of 2). Entries in parentheses indicate that the change has led to a non-integer half window width and therefore is unlikely to be employed.



	2	3	4	5	6	7	8
DIVIDE	46	91	68	91	76	91	80
MULTPLY	46	60	68	91	76	91	80
DOUBLE	46	—	92	—	—	—	138
HALVE	46	—	92	—	—	—	138
IHVALUE	91	91	91	91	91	91	19
CWINDOW (increasing)	12	24	36	48	60	163	84
CWINDOW (decreasing)	0	0	(82)	(237)	0	(237)	(237)
ZOOM (increasing)	52	76	96	139	151	163	154
ZOOM (decreasing)	40	52	(116)	(182)	66	(198)	(204)

Table: Relative Efforts

#### 4.9 Commands to Control Execution

EXIT	return to calling program	—	—	deactiv.
FORCE	recompute everything	—	—	activ
GO	compute whatever is needed	—	—	activ
QUIT	terminate MICROSCOPE session	—	—	—
RESTART	reestablish earlier parameter settings	—	—	—
SETDF	reset defaults	—	—	deactiv.
STORE	store current parameter settings	—	—	—
UNDO	undo previous parameter changes	—	—	—
WAIT	make program wait for GO or FORCE com.	—	—	deactiv.

To activate an inactive screen the GO or the FORCE command must be used. GO assumes that previous function evaluations are still valid and FORCE reevaluates the trial function at all needed points regardless of prior evaluations. Ordinarily GO will be used for the sake of efficiency, but the FORCE command is necessary if changes have been applied to the trial function (either after leaving MCRSCP or by using the USER command) that invalidate the earlier evaluations.

Usually, the command QUIT is used to terminate a MICROSCOPE session. That command leads to a simple FORTRAN STOP statement. However, it is possible to leave the routine MCRSCP via the EXIT command (leading to a RETURN statement), and then call MCRSCP again, perhaps after changing some screen parameters. After reentering MCRSCP, these will again be checked for consistency, and the screen will be marked inactive. However, the defaults will not be set, so that it is possible to return to MCRSCP and find everything as it has been left with the exception of what you may have reset yourself between MCRSCP calls. After reentering MCRSCP it may be necessary to use FORCE instead of GO to avoid peculiar displays due to altered screen parameters.

The recommended way of changing problem defining parameters (e.g. selecting a new trial function) is to use the command USER (see section 4.15).

Usually the screen is set inactive by some drastic change of key parameters such as the point of examination or the direction of investigation. It can also be set inactive by using the command WAIT. This option is particularly useful in the scrolling version when it may be convenient to effect several changes before scrolling the display just once, rather than having it scrolled after each individual command.

The state of a MICROSCOPE investigation can be stored using the command STORE, and later restored using the command RESTART. STORE writes the values of all relevant parameters onto the Restart Device and then rewinds that device. Thus, if the command STORE is given several times only the status during the last issue of the command can be restored. If it is necessary to store several data settings, then the restart channel number has to be changed by using the CCHANNEL command (see section 4.13), or a different file has to be assigned to it by the command USER.

If desired, some of the default settings can be reestablished by using the SETDF command. More precisely, the effects of SETDF are:

- 1- The screen is deactivated
- 2- All previous function evaluations are discarded
- 3- The window width  $w$  is set to 12
- 4- Only the display function itself is flagged to be plotted
- 5- Any cross derivatives are turned off
- 6- The discretization parameter  $h$  is set to its default value (1D-4)
- 7- The x-axis, center mark, and horizontal scale are all turned off

- 8- If the test package has been loaded (see Appendix III) then the test function is reset to its defaults ( $N = 1$ ,  $D = 10$ , no exponential is added).

While the screen is inactive, it may be possible to recover from an erroneous command using the command UNDO. This has an effect only on quantities set after the last GO or FORCE command, and resets them to their values prior to that command. The following parameters are reset (i.e. any change is undone):

- 1- The point of examination
- 2- The direction of investigation
- 3- The left and right endpoints of the investigation interval
- 4- The value of the discretization parameter  $h$
- 5- The window width  $w$
- 6- The number of independent variables
- 7- Marking the center point
- 8- Drawing a horizontal scale
- 9- Drawing a horizontal axis
- 10- Selection of graphs to be plotted

Note that any changes regarding a cross derivative or the normalization of the direction of investigation are not undone.

#### 4.10 Commands to Obtain On-Line Help

HELP	obtain detailed help on command cmd	cmd	2	—
HSUMRY	obtain a summary of all commands	—	—	—
LIST	print all currently available commands	—	—	—
NEWS	print the news	—	—	—

The quickest way of obtaining information on possible commands is the command LIST which will print a list of all currently available commands on the screen. More detailed information on all commands can be obtained through the HSUMRY command which prints the table in section 4.3 reorganized by topics.

Information on a specific command can be obtained by typing HELP, followed by the name of the command you want help on. MICROSCOPE will print not only a short description of the command, but also the current

values of those parameters that are affected by the command. Thus if, for example, you wish to find out what the current number of independent variables is you can request HELP on DMNSN.

The command NEWS will print a text that may change from time to time and will also depend on your particular installation. At the time of this writing the "news" are a very brief primer on MICROSCOPE.

Finally, if you type a question mark to MICROSCOPE, it will invite you to enter a command, and will suggest HSUMRY and LIST as possibilities.

#### 4.11 Commands to Keep a Record

LOG	give channel number for logging	nc	—	—
OUTPUT	write screen image on recording device	—	—	—
TNOTE	take a note on recording device	text	—	—

Results obtained with MICROSCOPE can be recorded permanently on the Record Device. The command TNOTE will copy any text you type on the Input Device until it encounters the string 'EC' (End Comments, both E and C must be capital) in columns 1 and 2. It then returns to command mode. The length of text lines may be up to 80 columns or the number of columns specified for the graphical display, whichever is larger. Trailing blanks are stripped, to make the record file more compact.

The command OUTPUT causes MICROSCOPE to write a copy of the graphic and the numerical display onto the recording device. This manual is generated by using TNOTE and OUTPUT throughout a lengthy string of MICROSCOPE commands.

There is also a logging device which is active if its channel number is different from zero (it is inactive by default). The command LOG sets the logging device number. If the device is active all commands, data, displays, and notes are recorded on it.

#### 4.12 Commands to Analyze Cross Derivatives

C1CROSS	read 1st component of cross direction	C1	C>0	deactiv.
C2CROSS	read 2nd component of cross direction	C2	C>0;1	deactiv.
C3CROSS	read 3rd component of cross direction	C3	C>0;1	deactiv.
CDIRCTN	read direction of cross direction	C	C>0	deactiv.
CHVALUE	read ch for cross differentiation	ch	5	deactiv.
CORDER	read order of cross derivative	k	-1<k<7	deactiv.
LCROSS	load a new cross direction	kc	3	deactiv.
RCROSS	read channel number for cross list	nc	—	—

Cross derivatives, like all derivatives in MICROSCOPE, are calculated using numerical differentiation. The formulas are analogous to those given for tangential derivatives. In general, the cross stencils will have no points in common. Therefore, one evaluation of a k-th order cross derivative requires k+1 evaluations of the trial function.

By choosing the cross direction to be parallel to the direction of investigation the range of tangential derivatives can be shifted from [0,6] to [k,6+k]. This situation, however, is not treated separately by MICROSCOPE and does not lead to savings in function evaluations.

The commands controlling the cross derivative (if any) are similar to those controlling the direction of investigation. By default, no cross derivative is computed. It is activated by reading a non-zero order of the derivative, using CORDER, or by defining a cross direction using CDIRCTN (which reads the cross direction), C1CROSS, C2CROSS, or C3CROSS (which read individual components), or LCROSS which loads a cross direction from a channel defined by RCROSS (the default channel number is zero). The default order is 1 if the cross derivative is activated by setting a cross direction. If it is activated by setting the order its default direction is the vector with all of its components equal to 1.

The cross discretization parameter ch is either set explicitly using CHVALUE, or by default, in which case it assumes the same value as the tangential parameter h at the time when the cross derivative is activated. For efficiency, later changes of h have no effect on the value of the cross parameter. Note that any change of ch requires a reevaluation of the display function at all needed points.

Cross derivatives are turned off by setting their order to zero (using CORDER). The NORMAL command has the same effect on cross derivatives as it does on tangential derivatives.

When plotting cross derivatives it is important to keep in mind that the significance of the window may be blurred if a stencil used for the cross differentiation, and centered outside of the window, nonetheless crosses the front. This can be avoided either by keeping the cross discretization parameter much smaller than the display interval, or by choosing the cross direction parallel to the front.

In the following example, the trial function is

$$f(x,y) = x^2 \cdot \text{abs}(x) \cdot y \cdot \text{abs}(y)$$

the point of examination is [0,1], the direction of investigation is [1,0], and the cross direction is [1,-1]. The value of h = 3D-3 and ch = 6D-3. Obviously the effects of the discontinuity in the second tangential derivative are carried beyond the window.

[illegible]

**Figure 27 An Extended Discontinuity**

#### 4.13 Programming MICROSCOPE

CCHANNEL	change channel number	ic,nc	4	—
PAUSE	pause until RETURN from channel nc	nc	—	—
RWIND	rewind device nc	nc	—	—

Some rudimentary programming of MICROSCOPE is possible. The basic approach consists of writing a sequence of commands into a file and assigning it to the input device. The various devices can be switched by using the CCHANNEL command. It will prompt for two integers, the first defining the type of channel to be switched, the second giving the new channel number. The types are

- 1: Input Device  
2: Output Device  
3: Graphics Device  
4: Recording Device  
5: Restart Device

For an explanation of these devices see section 1.3. Their channel numbers are defined originally by parameters passed on to the routine MCRSCP (see section 3.1). Upon entering MCRSCP these are copied into variables that may be changed by CCHANNEL. Thus the input parameters of MCRSCP are not changed by that command and assume their old values should MCRSCP be exited (using EXIT) and later reentered without changing its parameters.

The loading devices for the point of examination, the direction of investigation, and the cross direction can be switched by using the commands RPOINT, RDIRCTN and RCROSS, respectively. The Help Device cannot be switched as the information contained in it is an integral part of the MICROSCOPE package and is only read during a session. Indeed, it is advisable to protect the help file against an accidental overwrite if possible, e.g. by using a READONLY switch in an OPEN statement.

Any channel can be rewound by the RWIND command. This facility can be used e.g. to regenerate a specific parameter configuration or to overwrite previously generated information that is no longer needed. An infinite loop can be created by putting a rewind command at the end of the input file.

The PAUSE command can be used to view the output of a MICROSCOPE program at leisure during run time. It will ask for a channel number and then suspend computation until a dummy character is received from that channel. The channel will usually be the terminal.

Computation by a program can be suspended by changing the input device to the terminal (entering interactive mode) and later continued by switching the input channel back to the appropriate device.

Note that for all of the device handling commands no precaution has been taken to ensure that only legal channel numbers are given, and that appropriate devices are assigned to the channel numbers.

Comments can be included in a program in two ways. Firstly, a 'comment line' replaces a command line and is marked by a "+" character in column 1 or 2. Secondly, any characters contained in the space starting with column 3 of a command line are ignored by MICROSCOPE. This facility does not exist for lines containing data. Indeed, any spurious characters may lead to the rejection of the data by the parser.

#### 4.14 The PLOT command for obtaining a <PLOT79> Display

PLOT    enter plotting (<PLOT79>) mode        —    —    —

The alphanumerical displays usually generated by MICROSCOPE are adequate to obtain the desired answers to your questions, but they may not be suitable e.g. for publications, reports, etc. Depending upon the availability of suitable hard- and software (<PLOT79>, see Beebe, 1979) you may be able to generate more presentable plots. This is accomplished by the command PLOT which generates a drawing of the current alphanumerical display, augmented by some special options.

Details about the link between MICROSCOPE and <PLOT79> are given in section 5 and your system manager may have put out suitable local information in this respect. For the remainder of this subsection, we will assume that the <PLOT79> package is available, that you have linked it with MICROSCOPE and your own program, and that you have some suitable plotting device available. Appendix IV contains copies of black and white pen plots corresponding to the alphanumerical plots scattered throughout this manual.

The command PLOT puts you into the plot submode. The precise kind of plot you will eventually generate depends upon the MICROSCOPE commands given so far, and the specifications you are going to make in plot mode.

We first discuss the PLOT specifications. After entering the PLOT submode you will see a list of current options on the screen. You can change these options one by one. After each change an updated list will be displayed. To enter an option change enter the reference number of the option followed by the appropriate action (as prompted by the program), to activate the actual plotting process enter 0, and to leave the plotting mode and return to command mode without plotting anything enter any negative number. The list of PLOT options is left unchanged between calls to PLOT.

There are 12 PLOT options:

-1- Title (Default: blank)

A title printed above the alphanumerical display, consisting of up to 72 characters can be drawn.

-2- Legend (Default: blank)

A string of text printed below the alphanumerical display, consisting of up to 72 characters can also be drawn.



For both the title and the legend, a default character height is provided which will be decreased automatically if insufficient space is available.

-3- Labeling of graphs (Default: ON)

If ON, labels are drawn giving the degree of the derivative of each graph. On a crowded display, these are sometimes awkwardly placed, and marking the graphs (option 4) may be preferable.

-4- Marking of Graphs (Default: OFF)

If ON, each point at which a derivative has been evaluated is marked with a special symbol, similarly to the graphical display on the terminal. The symbols for the k-th derivative are

k:	0	1	2	3	4	5	6
Symbol:	dot	plus	asterisk	circle	x	square	triangle

Marking the graphs instead of labeling them has the advantage that it is possible to tell from any small part of the graph to which derivative it corresponds, making it unnecessary to trace a graph from the point of interest to its label. It has the disadvantage that it slows down significantly the speed with which the graphs can be drawn. Also, accentuating a graph is only possible with the marking on (see below). The marking symbol is repeated in the numerical display if the marking option is on, enabling easy association of marks and degrees of tangential derivatives.

-5- Drawing the plot in color (Default: ON)

The graph of the tangential derivatives can each be drawn in a different color, if supported by the existing hardware. In the most extreme case, when all derivatives are to be plotted in different colors, seven pens for their graphs plus an eighth pen for the numerical display, scales, axes, etc. must be supplied.

MICROSCOPE selects a pen by calling a routine SETPEN(I) where  $I = 1$  corresponds to the pen drawing the axes etc., and  $I > 1$  corresponds to the graph of the  $(I-2)$ th tangential derivative ( $I = 2, 3, \dots, 8$ ). This range of  $I$  has to be translated into the appropriate <PLOT79> range. On some installations the pens are numbered 0 through 7, on others from 1 to 8. The appropriate SETPEN routines are contained in the files PEN07 and PEN18, respectively (see section 5). If only a smaller range of pens is available you may want to write your own pen selection routine, tailoring it perhaps to the specific range of derivatives in which you are interested. Your routine must be of the following form:

```
SUBROUTINE SETPEN(I)
  INTEGER I,J
  C I is in the range from 1 to 8
    J = some function of I
  C J is in the range of pens available
  C on the local plotting device
  CALL STPEN(J)
  RETURN
END
```

The subroutine STPEN is a MICROSCOPE routine that calls the <PLOT79> routine SETPN(J) if the value of J has changed since the last call to STPEN. SETPEN is not referenced if the color option is turned off.

In the numerical display, the ranges of the tangential derivatives are drawn in the same colors as their graphs.

Using color makes for very nice drawings, but for reproduction purposes it may be preferable to use black and white (like the <PLOT79> drawings in Appendix IV).

-6- Drawing a page number (Default: None)

A large and prominent integer can be drawn below the legend. We have found this useful, for example, for associating a drawing with the reference number of a point in a list (e.g. the point of examination), and associating or defining a figure number (like in this manual).

No page number is drawn if its value is zero.

-7- Left bottom label (Default: none)

A label of up to 20 characters of small type can be drawn in the left bottom corner of the plot.

-8- Right bottom label (Default: none)

A similar label can be drawn in the right bottom corner of the plot.

-9- Time (Default: off)

The time at which the plot was generated can be drawn in the top left corner of the plot.

-10- Date (Default: OFF)

Similarly, the date of the plot generation can be drawn in the top right corner.

Both time and date are printed in the same size as the bottom labels. They are sometimes useful for identification and record keeping.

-11- Numerical Display (Default: ON)

If desired, the numerical display can be turned off. This is sometimes useful to obtain quick drawings.

-12- Frame (Default: ON)

If desired, the frame that is normally drawn around the entire <PLOT79> drawing can be turned off.

We now turn to the MICROSCOPE options. Essentially everything that appears in the alphanumerical screen display is also contained in the <PLOT79> drawing, although it may assume a slightly different form.

-1- Horizontal scale

If a horizontal scale is activated (by the DSCALE command) then a standard labeled horizontal scale is drawn on the bottom of the graphical <PLOT79> drawing. The labels on the scale give the distance from the point of examination in units of the direction of investigation. Notice that this may be the normalized version of the direction given in the numerical display, depending on whether the normalization option is on. If it is, then the scale in the <PLOT79> drawing gives the Euclidean distance from the point of examination.

-2- x Axis

A horizontal line is drawn through the center of the graphical display if the x axis is turned on (using DXAXIS). This is sometimes useful for illustrating symmetry properties.

-3- Marking the center of the display

If this option is turned on (using DCENTER) the center of the graphical display is marked with a cross.

-4- Accentuating a graph

If a graph is to be accentuated its marks (as discussed above) are drawn twice their normal size. However, if color is available, a better emphasis can usually be obtained by using a special color.

#### 4.15 User Intervention

USER     enter user mode

Upon receiving the command USER, MICROSCOPE calls a subroutine SUBUSR which is often a dummy routine, but which can be written to suit special purposes. SUBUSR has no parameters, but it can communicate with MICROSCOPE routines or your own routines via COMMON blocks. In utilizing this facility it is important to keep in mind that ordinarily MICROSCOPE has no means of determining that the trial function has changed and that previous evaluations of it are no longer valid. Therefore the command FORCE may have to be used rather than GO.

Five examples of possible applications follow:

##### -1- Mixed partial derivatives

If derivatives other than pure tangential derivatives of pure cross derivatives are to be examined this can be accomplished by defining the trial function to be the derivative that is to be displayed, or to be further differentiated, and make it dependent upon parameters that are modified by SUBUSR as needed. In this context, it may be convenient to use the routine

```
DOUBLE PRECISION FUNCTION DERIV(F,NDIM,P,IORDER,DIR,H)
DOUBLE PRECISION F,P(1),DIR(1),H
INTEGER NDIM,IORDER
...
RETURN
END
```

which computes at the point P the IORDERth derivative in the direction DIR of the double precision function F, using the formulas given in section 1.2 with the discretization parameter H. F must be declared EXTERNAL in the calling program, and is of the same form as the trial function described in section 3.1. NDIM is the number of independent variables F depends on and must not exceed 3. DERIV is used within MICROSCOPE (to compute any cross derivatives) and can therefore be readily called by your own programs.

Obviously, the modified trial function may itself be a tangential derivative. Then the range of derivatives that can be displayed is shifted from the normal 0th through 6th order to a higher one.

##### -2- Switching Trial Functions

If several trial functions are to be investigated in a single session (or with a single linking and loading process) they can all be written into one DOUBLE PRECISION FUNCTION, and distinguished by a

parameter that is passed in a common block, and that may be altered by SUBUSR. The examples in this manual are handled in this manner (see appendix III for details).

### -3- Switching Data or Parameters

Often the trial function depends upon data or parameters that can be changed by SUBUSR without having to leave MCRSCP.

### -4- File Handling

File handling is system dependent. You may find it convenient to handle the assignments of files or devices to channel numbers via USER.

#### 4.16 Default settings

In this subsection, E denotes a vector of 1 to 3 components that are all equal to 1. The default settings for MICROSCOPE are as follows:

Number of independent variables: 2

Point of examination: 0 (the origin)

Direction of Investigation: E

Discretization parameter: 1D-4

Window width: 12

Cross direction: E

Order of Cross Derivative: 0 (no cross derivative)

Order of Tangential Derivatives to be displayed: 0

No scale or x axis is drawn, the center of the display is unmarked, and no derivative is accentuated.

PLOT options: see section 4.14

## 5. Installation Guide

### 5.1 Description of the Package

In this section, which is directed mainly to the local system manager, we describe the files supplied with the MICROSCOPE package, and how they can be put together. Usually, it will be the best procedure to select or write files that generate the most powerful and convenient local version of MICROSCOPE, to put them into one file or a library and to make the appropriate information available to your local users.

MICROSCOPE is a portable set of FORTRAN routines. The user must supply a main program and one or two subroutines that are specific to the application. Thus users must have access to at least the relocatable versions of a suitable subset of MICROSCOPE, if not the source codes.

You should have obtained a magnetic tape containing 21 files in the following sequence:

- |     |         |        |
|-----|---------|--------|
| 1.  | HELPLC  | (902)  |
| 2.  | HELPUCL | (902)  |
| 3.  | LC      | (1889) |
| 4.  | MAIN    | (48)   |
| 5.  | MANLC   | (5150) |
| 6.  | MANUAL  | (7456) |
| 7.  | MANUC   | (5150) |
| 8.  | PEN07   | (9)    |
| 9.  | PEN18   | (8)    |
| 10. | PENDMY  | (3)    |
| 11. | PLTDMY  | (189)  |
| 12. | PLTLC   | (978)  |
| 13. | PLTUC   | (980)  |
| 14. | POS     | (14)   |
| 15. | SCREEN  | (89)   |
| 16. | SCROLL  | (86)   |
| 17. | SUPPRT  | (3894) |
| 18. | UC      | (1862) |
| 19. | USRDMY  | (8)    |
| 20. | USRLC   | (271)  |
| 21. | USRUC   | (270)  |

Numbers in parentheses give the number of lines (of up to 80 characters each) in the files.

Below, the files are divided into several groups. One or two files have to be picked from each of the groups 1 through 6. Within a group, files are listed by decreasing convenience, and increasing portability. If the bottom files are chosen from each group a completely

portable version of MICROSCOPE (except for the use of a few non-FORTRAN characters, namely ?, <, >, !, :, ; ) is obtained.

#### -1- Case Dependent Support Routines

LC: Contains those subroutines that require the support of lower case letters in order to work properly. This feature is not portable, but widely available, and greatly increases the convenience in using MICROSCOPE.

UC: To be used in case of LC, if lower case letters are not available. UC is supplied as part of the package, but it can also be generated from LC by replacing all lower case letters with their upper case equivalents, and by replacing the subroutine LCUC by a dummy routine. (Upper case versions of all other files can be generated from their lower case counter parts by replacing lower case letters by uppercase ones everywhere. The leading comments describing the file will of course no longer apply.)

If UC is used in case of LC all output will be in upper case letters, and lower case input will not be recognized.

LC and UC contain their respective versions of the master routine MCRSCP, which is the first routines in each of the two files. All other routines then follow alphabetically.

#### -2- Case Independent Routines

SUPPRT: This file contains the bulk of the MICROSCOPE package, consisting of completely portable routines.

#### -3- The File Access Routine

POS: This file contains a

SUBROUTINE POS(IDEV,N)

which positions the device IDEV at the N-th record, such that subsequent reading from IDEV yields the N-th record. This is accomplished by rewinding IDEV and then reading a dummy character from the first N-1 lines (records) in IDEV. If this at all time consuming it may be worthwhile to replace POS by a faster system dependent routine.

#### -4- The Screen Display Package

SCREEN: To be used if screen editing is locally available, and highly recommended over its alternative, SCROLL, described below. To take advantage of screen editing, the following four small interface routines

must be written for each possible computer/terminal combination (in all cases, IDEVCE is the channel number of the relevant terminal):

```

      SUBROUTINE CLRSCRN(IDEVCE,I,J)
      INTEGER IDEVCE,I,J
      C Clear the screen of the device with number IDEVCE below the point with
      C the coordinates (I,J) where I is the horizontal coordinate measured
      C from the left of the screen starting at 1, and J is the vertical
      C coordinate measured from the top of the screen starting at 1.
      ...
      RETURN
      END

      SUBROUTINE PCURSR(IDEVCE,I,J)
      INTEGER IDEVCE,I,J
      C Position the cursor at the position (I,J) of the device with number
      C IDEVCE, where I,J are as for CLRSCRN
      ...
      RETURN
      END

      SUBROUTINE PLCHRS(IDEVCE,I,J,N,CHARS)
      INTEGER IDEVCE,I,J,N,CHARS(135)
      C Write the string of N characters contained in A1 format in the integer
      C array CHARS, starting at the location (I,J). I,J, and IDEVCE are as
      C in CLRSCRN.
      ...
      RETURN
      END

      SUBROUTINE BLSCRN(IDEVCE)
      C Blank the screen of the device with number IDEVCE
      ...
      RETURN
      END
  
```

SCROLL: SCREEN can be replaced with SCROLL if the SCREEN package cannot be used. This will have the consequence that the entire screen display is regenerated after each change, rather than just modified.

#### -5- The PLOT package:

PLTLC: This file contains the interface between MICROSCOPE and <PLOT79>. The routines use and recognize lower case letters.

PLTUC: The equivalent of PLTLC, but without the lower case letters. Note, however, that <PLOT79> supports Hershey Fonts that include lower case letters regardless of the computing system. Hence the <PLOT7



drawings generated by PLTUC do contain lower case letters. The communication with the terminal via the command PLOT is completely in terms of upper case letters, however.

With both the PLTLC or PLTUC package a pen selection routine also has to be supplied (see section 4.14). The MICROSCOPE package contains three routines in separate files: PEN18 corresponding to 8 pens labeled 1 to 8, PEN07 corresponding to 8 pens labeled 0 to 7, and PENDMY to be used if color is not available (e.g. on dot matrix printer). However, you may wish to leave it up to the user to select or write his own pen selection routines.

PLTDMY: To be used if <PLOT79> is not available. In this case only alphanumerical screen displays like those shown throughout this manual can be obtained. However, the dummy routine simulates the interaction with <PLOT79> up to the point at which the actual drawing takes place. This feature is useful for the development of MICROSCOPE programs, and makes it possible to run programs without <PLOT79> even if they call the package.

There is only the upper case version of PLTDMY.

#### -6- The Test package

Ordinarily, users will not be interested in the specific routine SUBUSR employed in the generation of this manual. They will want to write their own. However, the manual version of SUBUSR, and the routines called by it, are contained in the files:

USRLC: contains the lower case version.

USRUC: contains the upper case version, to be used if lower case letters are not available.

USRDMY: contains a dummy SUBROUTINE SUBUSR, which disables the USER command. If the MICROSCOPE routines are held in a library it may be possible to keep this dummy routine in it as a standby if the user does not supply his own version.

#### -7- The <PLOT79> package

Obviously the <PLOT79> package must be linked with MICROSCOPE if it is to be utilized. This is a large graphics package that can be easily interfaced with almost any graphics device. It can be obtained for a nominal charge from Dr. Nelson H.F. Beebe, Department of Physics, University of Utah, Salt Lake City, Utah 84112, Tel.: 801-581-5254.

#### -8- Data files

#### Sample Driver Program

The file MAIN contains the driver program as it was used in the generation of this manual. It is listed as a data file because the program needs to be modified to suit local circumstances. In particular, the OPEN statement is not portable.

#### Interactive Help Files

The package includes two help files that contain relevant help information (but no code). Again, there are the lower and upper case versions:

HELPLC

HELPU

One of these files (preferably HELPLC) must be readable by the user programs, and the users must be informed how to access them. It is possible to design a sequence of MICROSCOPE commands that overwrites the help file, and, if possible, the user should be denied write access to it. You may also want to keep easily accessible backup copies of these files.

The help files each contain the following items:

1. On the first line there are three integers I,M,N in I3 format. These have the following meanings: I is the number of lines occupied by the table of commands that can be accessed by the command HSUMRY. M is the total number of lines occupied by the detailed information accessed by HELP. N is the number of available commands.
2. The next four lines are occupied by N numbers written in 2012 format. The k-th such number gives the length (in lines) of the detailed HELP information on the k-th command. These N numbers must add up to M.
3. The next I lines contain the HSUMRY information
4. The next M lines contain the detailed HELP information.
5. The next line contains an integer J in I3 format that contains the number of lines occupied by the news.
6. The last J lines contain the news.

#### Manual files

MANUAL: contains a verbatim machine readable copy of this manual. It is written with FORTRAN carriage control and ready for printing (at 60 lines per page and up to 75 characters per line). Underlining is accomplished by having the underlines in a separate line with a "+" character in the first column, indicating that the line is to overwrite the

preceding one. There is no upper case version of this file. If lower case letters are not available a xeroxed copy of the manual rather than a printed version should be distributed.

This installation guide is chapter five of the manual.

MANLC: This file contains a version of the manual that is suitable for on-line reading and editing. Page breaks and underlines have been eliminated, otherwise the text is identical to that in MANUAL.

MANUC: The upper case equivalent of MANLC.

### 5.2 A Key to Selecting the Appropriate Files

The following is a quick key to selecting the appropriate subset of the above files. Proceed by answering the questions and following instructions until you reach a line giving the relevant file names or descriptions. The pen selection routine is described in section 4.14 (and can be supplied by the user if necessary), all other ingredients are described in this installation guide.

- |                                                                                                                                       |                             |
|---------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| 1. Are lower case letters available on your machine?                                                                                  | Yes GO TO 2<br>No GO TO 3   |
| 2. Have you supplied an interface to screen editing?                                                                                  | Yes GO TO 4<br>No GO TO 5   |
| 3. Have you supplied an interface to screen editing?                                                                                  | Yes GO TO 6<br>No GO TO 7   |
| 4. Is <PLOT79> available?                                                                                                             | Yes GO TO 8<br>No GO TO 9   |
| 5. Is <PLOT79> available?                                                                                                             | Yes GO TO 10<br>No GO TO 11 |
| 6. Is <PLOT79> available?                                                                                                             | Yes GO TO 12<br>No GO TO 13 |
| 7. Is <PLOT79> available?                                                                                                             | Yes GO TO 14<br>No GO TO 15 |
| 8. LC, SUPPRT, POS or local equivalent, SCREEN, PLTLC, pen selection routine, possibly USRLC, screen interface, HELPLC, MANUAL, MANLC |                             |
| 9. LC, SUPPRT, POS or local equivalent, SCREEN, PLTDMY, possibly USRLC, screen interface, HELPLC, MANUAL, MANLC                       |                             |

10. LC, SUPPRT, POS or local equivalent, SCROLL, PLTLC, pen selection routine, possibly USRLC, HELPLC, MANUAL, MANLC
11. LC, SUPPRT, POS or local equivalent, SCROLL, PLTDMY, possibly USRLC, HELPLC, MANUAL, MANLC
12. UC, SUPPRT, POS or local equivalent, SCREEN, PLTUC, pen selection routine, possibly USRUC, screen interface, HELPUC, MANUC
13. UC, SUPPRT, POS or local equivalent, SCREEN, PLTDMY, possibly USRUC, screen interface, HELPUC, MANUC
14. UC, SUPPRT, POS or local equivalent, SCROLL, PLTUC, pen selection routine, possibly USRUC, HELPUC, MANUC
15. UC, SUPPRT, POS or local equivalent, SCROLL, PLTDMY, possibly USRUC, HELPUC, MANUC

### Appendix I: Numerical Differentiation

In this appendix, we will discuss in some detail the properties of the formulas listed in section 1.2. Without loss of generality we assume the point of examination is the origin. We also restrict our attention to purely tangential derivatives. Thus we may assume there is only one independent variable. Of particular interest are step discontinuities in certain derivatives (for other types of reduced smoothness see the examples at the end of Appendix III). Thus we consider the trial function

$$F(x) = \text{PHI}(x) + S(m, \text{eta}, x)$$

where  $\text{PHI}(x)$  is a smooth function (i.e. all its relevant derivatives are continuous) and  $S$  is determined by the requirements that its  $m$ -th derivative be piecewise constant and have a jump discontinuity of magnitude  $\text{eta}$  at  $x = 0$ . Explicitly,  $S$  is defined by

$$S(m, \text{eta}, x) = 0 \text{ if } x < 0 \text{ and } \text{eta} x^m / m! \text{ otherwise}$$

where the exclamation mark denotes the factorial.

To simplify matters we will assume  $S$  is evaluated exactly, and all round-off errors occur in  $\text{PHI}$ . The properties of the numerical differentiation formulas depend very critically on the relative sizes of the derivatives of  $\text{PHI}$ , about which little can be assumed in general. Our approach will be to derive general formulas incorporating the derivatives, but to illustrate the relevant phenomena by using

$$\text{PHI}(x) = e^x$$

since the derivatives of the exponential are all identical (assuming the normalization option is on).

We will denote the degree of a derivative by parenthesized superscripts.

A simple calculation shows that, for  $k \geq m$ , the approximation  $Q^{(k)}$  of  $f^{(k)}$  at  $x = 0$  is given by the fundamental equation

$$\begin{aligned}
 Q^{(k)} &= \text{PHI}^{(k)} && \text{true value} \\
 &+ h \cdot \gamma_k \cdot \text{PHI}^{(k+2)} && \text{truncation error} \\
 &+ \tau \cdot h^{-k} \cdot (1 + \text{abs}(\text{PHI})) \cdot \alpha_k && \text{round-off term} \\
 &+ \epsilon \cdot \delta_{m,k} && \text{delta term}
 \end{aligned} \tag{1}$$

If  $k < m$  the delta term is absent. We discuss the individual terms in turn:

- The true value

This is the expression which we are trying to approximate.  $\text{PHI}^{(k)}$  is evaluated at zero.

- The truncation error.

This term can be obtained by expanding the error into a Taylor series about  $h = 0$ , as is done in any textbook on numerical analysis. The function  $\text{PHI}^{(k+2)}$  is to be thought of as having been evaluated at a suitable (unknown) point in the domain of the display. The  $\gamma_k$  constants are given in table 1.

- The round-off term

We assume rounding is taking place as described in section 0, i.e. if  $x$  is the quantity to be rounded to  $D$  digits, say, then  $x$  is replaced by

$$z := (1 + \epsilon_1 \cdot 10^{-D}) \cdot x + \epsilon_2 \cdot 10^{-D} \tag{2}$$

where  $\epsilon_1$  and  $\epsilon_2$  are random numbers between  $-1$  and  $+1$ . The quantity  $\tau$  in (1) is a suitable average of the  $\epsilon_1$  and  $\epsilon_2$  terms in (2). We will

-D  
set it to  $10^{-k}$ . The function PHI is again evaluated at a suitable point in the display, and the  $\alpha_k$  are the sum of the absolute values of the coefficients in the k-th differentiation formula. Their values are listed in table 1.

- The delta term.

We use the expression "delta term" because it arises from differentiating the step function  $s(m, \eta, x)$ , and the first derivative of a step function is a Dirac delta function. The  $\delta_{m,k}$  are constants that can be read off the MICROSCOPE plots in section 4.2. They are also given in table 1.

k	$\alpha_k$	$\gamma_k$	delta						
			$\delta_{m,k}$						
			m = 0	1	2	3	4	5	6
1	2	1/6	1/2	1	-	-	-	-	-
2	4	1/12	1	1	1	-	-	-	-
3	24	1/16	4	2	1	1	-	-	-
4	256	1/24	48	16	3	1	1	-	-
5	2430	1/27	365	81	14	3	1	1	-
6	46656	1/36	7290	1458	146	27	4	2	1

Table 1: Constants for Differentiation Formulas

We now address three questions:

1. When do round-off effects become visible?
2. When do round-off effects exceed the truncation error?
3. How small a discontinuity can be detected with MICROSCOPE?

### 1.1 The Round-Off Threshold

We are concerned with the question of when round-off effects begin to obscure the graph of a derivative. More precisely, we ask when the round-off term becomes larger than a certain fraction of the rise in  $\Phi^{(k)}$

the graph of the derivative  $\Phi^{(k)}$  over an interval of length  $s$ . An equation for the critical step size, which constitutes a limit on  $h$  below which no clear graphs can be obtained, is generated by ignoring the delta  $\Phi^{(k)}$

term and the truncation error, and approximating the rise of  $\Phi^{(k)}$  by the mean value theorem. This leads to:

$$\psi_1 h^{-k} (1 + \text{abs}(\Phi^{(k)})) \tau \alpha = 2 h \text{abs}(\Phi^{(k+1)}) / w$$

where  $w = 2h/s$  is the window width,  $\Phi^{(k+1)}$  again is evaluated at a suitable point, and  $\psi_1$  is a safety factor. The larger  $\psi_1$ , the clearer will be the graph of  $\Phi^{(k)}$ .

Solving for  $h$  yields:

$$h = [\psi_1 (1 + \text{abs}(\Phi^{(k)})) \tau \alpha w / (2 \text{abs}(\Phi^{(k+1)}))]^{1/(k+1)}$$

Table 2 contains threshold values for various values of  $D$  (the number of digits carried) and  $\psi_1 = 5$ , for the case that  $\Phi^{(k)}$  is the exponential. All derivatives have been evaluated at  $x = 0$ .



Digits (D)	Degree of Derivative (k)					
	1	2	3	4	5	6
1	3.46D+00	2.88D+00	3.46D+00	4.34D+00	4.94D+00	6.00D+00
2	1.10D+00	1.34D+00	1.95D+00	2.74D+00	3.37D+00	4.32D+00
3	3.46D-01	6.21D-01	1.10D+00	1.73D+00	2.29D+00	3.11D+00
4	1.10D-01	2.88D-01	6.16D-01	1.09D+00	1.56D+00	2.24D+00
5	3.46D-02	1.34D-01	3.46D-01	6.88D-01	1.06D+00	1.61D+00
6	1.10D-02	6.21D-02	1.95D-01	4.34D-01	7.25D-01	1.16D+00
7	3.46D-03	2.88D-02	1.10D-01	2.74D-01	4.94D-01	8.34D-01
8	1.10D-03	1.34D-02	6.16D-02	1.73D-01	3.37D-01	6.00D-01
9	3.46D-04	6.21D-03	3.46D-02	1.09D-01	2.29D-01	4.32D-01
10	1.10D-04	2.88D-03	1.95D-02	6.88D-02	1.56D-01	3.11D-01
11	3.46D-05	1.34D-03	1.10D-02	4.34D-02	1.06D-01	2.24D-01
12	1.10D-05	6.21D-04	6.16D-03	2.74D-02	7.25D-02	1.61D-01
13	3.46D-06	2.88D-04	3.46D-03	1.73D-02	4.94D-02	1.16D-01
14	1.10D-06	1.34D-04	1.95D-03	1.09D-02	3.37D-02	8.34D-02
15	3.46D-07	6.21D-05	1.10D-03	6.88D-03	2.29D-02	6.00D-02
16	1.10D-07	2.88D-05	6.16D-04	4.34D-03	1.56D-02	4.32D-02
17	3.46D-08	1.34D-05	3.46D-04	2.74D-03	1.06D-02	3.11D-02
18	1.10D-08	6.21D-06	1.95D-04	1.73D-03	7.25D-03	2.24D-02
19	3.46D-09	2.88D-06	1.10D-04	1.09D-03	4.94D-03	1.61D-02
20	1.10D-09	1.34D-06	6.16D-05	6.88D-04	3.37D-03	1.16D-02
21	3.46D-10	6.21D-07	3.46D-05	4.34D-04	2.29D-03	8.34D-03
22	1.10D-10	2.88D-07	1.95D-05	2.74D-04	1.56D-03	6.00D-03
23	3.46D-11	1.34D-07	1.10D-05	1.73D-04	1.06D-03	4.32D-03
24	1.10D-11	6.21D-08	6.16D-06	1.09D-04	7.25D-04	3.11D-03
25	3.46D-12	2.88D-08	3.46D-06	6.88D-05	4.94D-04	2.24D-03
26	1.10D-12	1.34D-08	1.95D-06	4.34D-05	3.37D-04	1.61D-03
27	3.46D-13	6.21D-09	1.10D-06	2.74D-05	2.29D-04	1.16D-03
28	1.10D-13	2.88D-09	6.16D-07	1.73D-05	1.56D-04	8.34D-04
29	3.46D-14	1.34D-09	3.46D-07	1.09D-05	1.06D-04	6.00D-04
30	1.10D-14	6.21D-10	1.95D-07	6.88D-06	7.25D-05	4.32D-04
31	3.46D-15	2.88D-10	1.10D-07	4.34D-06	4.94D-05	3.11D-04
32	1.10D-15	1.34D-10	6.16D-08	2.74D-06	3.37D-05	2.24D-04
33	3.46D-16	6.21D-11	3.46D-08	1.73D-06	2.29D-05	1.61D-04
34	1.10D-16	2.88D-11	1.95D-08	1.09D-06	1.56D-05	1.16D-04
35	3.46D-17	1.34D-11	1.10D-08	6.88D-07	1.06D-05	8.34D-05
36	1.10D-17	6.21D-12	6.16D-09	4.34D-07	7.25D-06	6.00D-05
37	3.46D-18	2.88D-12	3.46D-09	2.74D-07	4.94D-06	4.32D-05
38	1.10D-18	1.34D-12	1.95D-09	1.73D-07	3.37D-06	3.11D-05
39	3.46D-19	6.21D-13	1.10D-09	1.09D-07	2.29D-06	2.24D-05
40	1.10D-19	2.88D-13	6.16D-10	6.88D-08	1.56D-06	1.61D-05

Table 2: Guide to selecting a value of the discretization parameter

### 1.2 Round-off versus Truncation Errors

In working with MICROSCOPE, one will rarely be interested in obtaining accurate numerical values of derivatives. Rather, qualitative features will be of primary concern. However, if accuracy is paramount, then the discretization parameter should be picked as small as possible without the round-off term swamping the truncation error. The optimal step size is obtained by equating the round-off term with the truncation error and solving for h. This yields:

$$h = \left[ \frac{(1 + \alpha \beta(PHI)) \cdot \tau \cdot \alpha}{k} \cdot \frac{1}{(\gamma \cdot PHI)^{1/(k+2)}} \right]$$

Table 3 lists values of h for the exponential.

Digits (D)	Degree of Derivative (k)					
	1	2	3	4	5	6
1	1.69D+00	2.49D+00	4.50D+00	8.25D+00	1.18D+01	1.88D+01
2	7.83D-01	1.40D+00	2.84D+00	5.62D+00	8.49D+00	1.41D+01
3	3.63D-01	7.87D-01	1.79D+00	3.83D+00	6.11D+00	1.06D+01
4	1.69D-01	4.43D-01	1.13D+00	2.61D+00	4.40D+00	7.93D+00
5	7.83D-02	2.49D-01	7.13D-01	1.78D+00	3.17D+00	5.95D+00
6	3.63D-02	1.40D-01	4.50D-01	1.21D+00	2.28D+00	4.46D+00
7	1.69D-02	7.87D-02	2.84D-01	8.25D-01	1.64D+00	3.34D+00
8	7.83D-03	4.43D-02	1.79D-01	5.62D-01	1.18D+00	2.51D+00
9	3.63D-03	2.49D-02	1.13D-01	3.83D-01	8.49D-01	1.88D+00
10	1.69D-03	1.40D-02	7.13D-02	2.61D-01	6.11D-01	1.41D+00
11	7.83D-04	7.87D-03	4.50D-02	1.78D-01	4.40D-01	1.06D+00
12	3.63D-04	4.43D-03	2.84D-02	1.21D-01	3.17D-01	7.93D-01
13	1.69D-04	2.49D-03	1.79D-02	8.25D-02	2.28D-01	5.95D-01
14	7.83D-05	1.40D-03	1.13D-02	5.62D-02	1.64D-01	4.46D-01
15	3.63D-05	7.87D-04	7.13D-03	3.83D-02	1.18D-01	3.34D-01
16	1.69D-05	4.43D-04	4.50D-03	2.61D-02	8.49D-02	2.51D-01
17	7.83D-06	2.49D-04	2.84D-03	1.78D-02	6.11D-02	1.88D-01
18	3.63D-06	1.40D-04	1.79D-03	1.21D-02	4.40D-02	1.41D-01
19	1.69D-06	7.87D-05	1.13D-03	8.25D-03	3.17D-02	1.06D-01
20	7.83D-07	4.43D-05	7.13D-04	5.62D-03	2.28D-02	7.93D-02
21	3.63D-07	2.49D-05	4.50D-04	3.83D-03	1.64D-02	5.95D-02
22	1.69D-07	1.40D-05	2.84D-04	2.61D-03	1.18D-02	4.46D-02
23	7.83D-08	7.87D-06	1.79D-04	1.78D-03	8.49D-03	3.34D-02
24	3.63D-08	4.43D-06	1.13D-04	1.21D-03	6.11D-03	2.51D-02
25	1.69D-08	2.49D-06	7.13D-05	8.25D-04	4.40D-03	1.88D-02
26	7.83D-09	1.40D-06	4.50D-05	5.62D-04	3.17D-03	1.41D-02
27	3.63D-09	7.87D-07	2.84D-05	3.83D-04	2.28D-03	1.06D-02
28	1.69D-09	4.43D-07	1.79D-05	2.61D-04	1.64D-03	7.93D-03
29	7.83D-10	2.49D-07	1.13D-05	1.78D-04	1.18D-03	5.95D-03
30	3.63D-10	1.40D-07	7.13D-06	1.21D-04	8.49D-04	4.46D-03
31	1.69D-10	7.87D-08	4.50D-06	8.25D-05	6.11D-04	3.34D-03
32	7.83D-11	4.43D-08	2.84D-06	5.62D-05	4.40D-04	2.51D-03
33	3.63D-11	2.49D-08	1.79D-06	3.83D-05	3.17D-04	1.88D-03
34	1.69D-11	1.40D-08	1.13D-06	2.61D-05	2.28D-04	1.41D-03
35	7.83D-12	7.87D-09	7.13D-07	1.78D-05	1.64D-04	1.06D-03
36	3.63D-12	4.43D-09	4.50D-07	1.21D-05	1.18D-04	7.93D-04
37	1.69D-12	2.49D-09	2.84D-07	8.25D-06	8.49D-05	5.95D-04
38	7.83D-13	1.40D-09	1.79D-07	5.62D-06	6.11D-05	4.46D-04
39	3.63D-13	7.87D-10	1.13D-07	3.83D-06	4.40D-05	3.34D-04
40	1.69D-13	4.43D-10	7.13D-08	2.61D-06	3.17D-05	2.51D-04

Table 3: Values of h for maximum accuracy

### 1.3 Derivatives of Step Functions

It turns out that very small jumps (of magnitude  $\epsilon$ , say) in some derivative can be detected only by examining a higher order derivative rather than the discontinuous derivative itself. We determine optimal values of  $h$  and minimum values of  $\epsilon$  by imposing two requirements. First, the delta term should be significantly larger than round-off errors, and, second, the delta term should be significantly larger than the rise of  $\text{PHI}^{(k)}$  over half the window. This leads to the equations

$$\epsilon \delta_{m,k} h^{m-k} = \psi_2 h^{-k} (1 + \text{abs}(\text{PHI})) \tau \alpha_k$$

and

$$\epsilon \delta_{m,k} h^{m-k} = \psi_3 h^{(k+1)} \text{PHI}$$

where  $\psi_2$  and  $\psi_3$  are safety factors.

We consider two cases. If  $m = 0$  (i.e. there is a step in the trial function) equation (4) is independent of  $h$ . Solving for  $\epsilon$  yields the minimum value

$$\epsilon_0 = \psi_2 (1 + \text{abs}(\text{PHI})) \tau \alpha_k / \delta_{0,k}$$

The maximum value of  $h$  (above which the delta term will be obscured by the growth in  $\text{PHI}^{(k)}$ ) is obtained by solving (5) for  $h$ :

$$h = [\epsilon \delta_{0,k} / (\psi_3 \text{PHI}^{(k+1)})]^{1/(k+1)}$$

If  $m > 0$  (i.e. the discontinuity occurs in a derivative) the minimum value of  $\epsilon$  can be computed by eliminating  $h$  between (4) and (5), and solving for  $\epsilon$ . This gives

$$\eta = \left[ \frac{(\psi \cdot \text{abs}(\text{PHI}))^{k+1-m}}{3} \right] \cdot \left[ \frac{(\psi \cdot (1 + \text{abs}(\text{PHI})) \cdot \tau \cdot \alpha)^{k+1-m}}{2} \right]^{1/(k+1)} / \delta m.$$

The corresponding optimal value of  $h$  (above which the delta term is too small to be visible in the general growth of  $\text{PHI}$  and below which it is obscured by round-off errors) can be obtained by eliminating  $\eta$  between (4) and (5) and solving for  $h$ , giving

$$h = \left( \frac{(\psi \cdot (1 + \text{abs}(\text{PHI})) \cdot \alpha \cdot \tau)}{2} \right)^{1/(k+1)} / \psi$$

Notice that this expression is independent of  $m$ , the degree of the derivative in which the jump occurs.

Table 4 lists minimum values of  $\eta$  (in the last seven columns) and the corresponding optimal value of  $h$  (in the second column) for the case of the exponential, and for a few values of  $\tau$ . Notice how the size of the jump that can be detected is sometimes decreased by several orders of magnitude if higher order derivatives are examined.

$\tau = 1.00-10$

		Degree of discontinuous derivative						
k	optimal h	0	1	2	3	4	5	6
0	5.00-10	1.50-09						
1	4.50-05	1.20-08	1.30-04					
2	1.60-03	1.20-08	7.60-06	4.80-03				
3	1.20-02	1.80-08	2.90-06	4.60-04	3.70-02			
4	4.80-02	1.60-08	1.00-06	1.10-04	6.90-03	1.40-01		
5	1.20-01	2.00-08	7.80-07	3.90-05	1.60-03	4.00-02	3.50-01	
6	2.40-01	1.90-08	4.00-07	1.70-05	3.70-04	1.00-02	8.70-02	7.20-01

$\tau = 1.0D-17$

		Degree of discontinuous derivative						
k	optimal h	0	1	2	3	4	5	6
0	5.0D-17	1.5D-16						
1	1.4D-08	1.2D-15	4.2D-08					
2	7.4D-06	1.2D-15	1.6D-10	2.2D-05				
3	2.2D-04	1.8D-15	1.6D-11	1.5D-07	6.6D-04			
4	1.9D-03	1.6D-15	2.5D-12	7.0D-09	1.1D-05	5.7D-03		
5	7.9D-03	2.0D-15	1.1D-12	8.3D-10	4.9D-07	1.9D-04	2.4D-02	
6	2.4D-02	1.9D-15	4.0D-13	1.7D-10	3.7D-08	1.0D-05	8.7D-04	7.2D-02

$\tau = 1.0D-24$

		Degree of discontinuous derivative						
k	optimal h	0	1	2	3	4	5	6
0	5.0D-24	1.5D-23						
1	4.5D-12	1.2D-22	1.3D-11					
2	3.4D-08	1.2D-22	3.5D-15	1.0D-07				
3	3.9D-06	1.8D-22	9.1D-17	4.6D-11	1.2D-05			
4	7.6D-05	1.6D-22	6.3D-18	4.4D-13	1.7D-08	2.3D-04		
5	5.4D-04	2.0D-22	1.7D-18	1.8D-14	1.6D-10	8.7D-07	1.6D-03	
6	2.4D-03	1.9D-22	4.0D-19	1.7D-15	3.7D-12	1.0D-08	8.7D-06	7.2D-03

$\tau = 1.0D-31$

		Degree of discontinuous derivative						
k	optimal h	0	1	2	3	4	5	6
0	5.0D-31	1.5D-30						
1	1.4D-15	1.2D-29	4.2D-15					
2	1.6D-10	1.2D-29	7.6D-20	4.8D-10				
3	7.0D-08	1.8D-29	5.1D-22	1.5D-14	2.1D-07			
4	3.0D-06	1.6D-29	1.6D-23	2.8D-17	2.8D-11	9.1D-06		
5	3.7D-05	2.0D-29	2.5D-24	3.9D-19	4.9D-14	4.0D-09	1.1D-04	
6	2.4D-04	1.9D-29	4.0D-25	1.7D-20	3.7D-16	1.0D-11	8.7D-08	7.2D-04

Table 4: The superiority of testing Delta Functions and their derivatives

## Appendix II: Organization of MICROSCOPE

This appendix contains (1) a brief discussion of the organizational framework upon which MICROSCOPE is built, (2) a detailed explanation of the sampling scheme and logic which MICROSCOPE uses to calculate the tangential derivatives, and (3) a brief description of the common blocks used.

MICROSCOPE has evolved over a period of several years, progressing through several preliminary versions. The core of the present version was designed and written by Bill Harris during the summer of 1983. MICROSCOPE then grew, under the hands of Peter Alfeld, to a substantially larger package which is the current version. This explains lack of uniformity in the commenting of the source code. It also accounts for the inconsistency in the notation used within the program versus that employed in this manual. However, only very serious users interested in understanding and possibly manipulating the code (for whom this appendix is written) should suffer from such idiosyncrasies of the package.

### II.1 Organization of the Program Package

MICROSCOPE follows a common and well established format for interactive programs in FORTRAN. The user must provide a short main program, which assigns FORTRAN Input/Output device numbers and screen parameters, and then calls a master routine (MCRSCP). MCRSCP is a driver program from which all other routines are called and to which execution returns after each command has been completed. This is accomplished by (i) prompting for a new command, (ii) reading and recognizing the new command, (iii) using a computed GO TO to branch to the appropriate routine, (iv) a bit of logic and computation (i.e. sampling the function and calculating the derivatives), (v) displaying the results in some graphic form, and (vi) updating an options table and returning to step (i).

To help understanding the following flowchart several ideas and their representative variables need to be introduced. The logical variable LGO indicates whether the screen is active (LGO = .TRUE) or inactive (LGO = .FALSE.). It is set and reset at a number of places in the package. The only place where the value of LGO has an effect on the program flow is in the routine MCRSCP, between statements 7900 and 8300.

There are 4 basic types of commands. These correspond to the integer variable ITYPE which is set appropriately by each command. Commands of ITYPE = 1 imply that the function must be sampled from scratch. Since other changes may be desired before computation begins, these commands deactivate the screen and this is accomplished by setting LGO = .FALSE.. Commands which are of ITYPE = 2 require some computation. Two special ITYPE = 2 commands are FORCE and GO. These cause LGO to be set to .TRUE. and are the only commands that activate an inactive screen. The

remaining commands of ITYPE = 2 only make modifications of the current setup and usually only require small amounts of additional computation. When LGO = .TRUE. any ITYPE = 2 command is executed before the next command is prompted for. ITYPE = 3 commands do not require computation but cause the alphanumerical display to be altered. Finally, commands of ITYPE = 4 do not alter the display and do not require computation.

The other major logical control variable is LSCRN. Its main use is in screen-editing versions of MICROSCOPE. LSCRN = .FALSE. means that the alphanumerical display has been overwritten by text and must be regenerated. LSCRN = .TRUE. means that the display is intact from the previous command and only selected pixels need to be updated. LSCRN is used by the display routine SGRAPH in the file SCREEN.

Computation time for evaluating display functions can be sizable. The program logic and sampling scheme were designed to utilize previous computations and eliminate unnecessary recomputations to the extent possible. For example, shifts and magnification changes often require only a few additional evaluations.

The flowchart that follows gives an overview of the logical flow in the MCRSCP driver routine. It concentrates on the portion between statement 1500 and 8300 where the logic is somewhat involved.



Epitome Flowchart

```

!
! from Main Program
!
V
• output message identifying MICROSCOPE
• check screen parameters
• copy channel numbers, screen parameters, and set flags
• set LGO = .FALSE.
• compute round-off characteristics (NUMDIG)
• set default values for some of the variables contained in
  the various common blocks
• input some of the help documentation as well as news (INHELP)
1500 • prompt/input/parse command (DIALOG)
• reset error flags
• branch to the appropriate subroutine (using a computed GO TO)
  This corresponds to statements 1600 through 7800. Many
  of these commands alter variables, with subscript 2, in
  the common block OPTION. These will then be used by
  later routines.
7900 • if command is of ITYPE = 1 (i.e. requires complete recomputation)
  then set: LGO = .FALSE.
• if ITYPE = 2 and LGO = .TRUE.
  then calculate values for NUM and DEN corresponding to
  sampling changes. (See documentation in part 2
  of this appendix) (SGAMMA)
• check for errors and output appropriate messages if any occurred
• if error = .TRUE.
  <----- then go to 1500
• if LGO = .FALSE. and the Refresh Screen (RSCREEN) command is given
  then display the old graphical information (SGRAPH)
• if LGO = .FALSE.
  <----- then go to 1500
• branch to the appropriate section of the program depending on
  the value of ITYPE (using a computed GO TO)
8000 • (if ITYPE = 2)
• check to see which aspects of the sampling have been altered
  by the command and set the appropriate logical flags (CHKCMP)
• call the routines flagged by the CHKCMP routine.
  (the actual sampling and computation of derivatives occurs
  only in the routine SCMUPD)
8100 • (if ITYPE = 3)
• set up the graphics data (i.e. update plot array) (SGRUPD)
• graph the data (SGRAPH)
• if logging is required
  then output a copy of the graphical display in scroll mode
• copy current set of parameter values from new to old. (OPCOPY)
  <----- go to 1500

```

## II.2 Sampling Scheme and Logic

In the remainder of this appendix an explanation of the sampling scheme and computational logic is given. The variable *h*, as used in the source code, is not the same as the variable labeled as *h* on output. To reduce confusion we will use the following notation in this appendix (but nowhere else in this manual):

Variable in Source Code	Variable in Appendix	Variable in Output and elsewhere in the manual
<i>h</i>	<i>ss</i> (step size)	<i>s</i>
<i>sw</i>	<i>sw</i> (stencil width)	<i>h</i>

The output names correspond to the common usage while the names used in the program and appendix have the meanings defined below.

The basic idea of the sampling scheme consists of representing the relevant part of the line of investigation in two linear arrays, one corresponding to points in the domain, and one containing corresponding values of the display function (if it has been evaluated). Any changes in the stencil width and step size, as well as small shifts of the point of examination are then implemented by shuffling the information in these two arrays. The size of the arrays has been chosen to accomodate fairly large changes in parameter values with little computation. The key variables and arrays used in the program are as follows:

AD-A144 720

MICROSCOPE: A SOFTWARE SYSTEM FOR MULTIVARIATE ANALYSIS  
(U) WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER  
P ALFELD ET AL. JUN 84 MRC-TSR-2701 DAAG29-80-C-0041

2/2

UNCLASSIFIED

F/G 9/2

NL

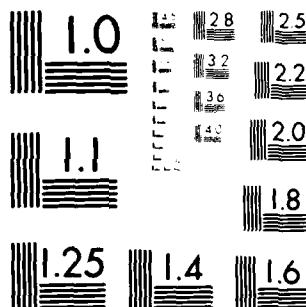
END

DATE

FORMED

9-84

DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

XS(IXSIZE): the sampling array for the domain space, giving the distance from the point of examination in units of the direction of investigation.

FS(IXSIZE): the sampling array containing the value of the display function  $f$  at the corresponding point in XS().

IXSIZE: the dimension of the XS() and FS() arrays. (the current value of IXSIZE is 5377)

ICENTR: the index of the center element of the XS() and FS() arrays. (the current value of ICENTR is 2689)

ROPSTS(2): this is the "step size" corresponding to 8 points in the XS() array. (see the illustration below) It appears in the common block OPTION. The reason for introducing it here is that for most computations it will remain fixed while the other parameters, such as sw and ss, will be doubled, halved, etc.

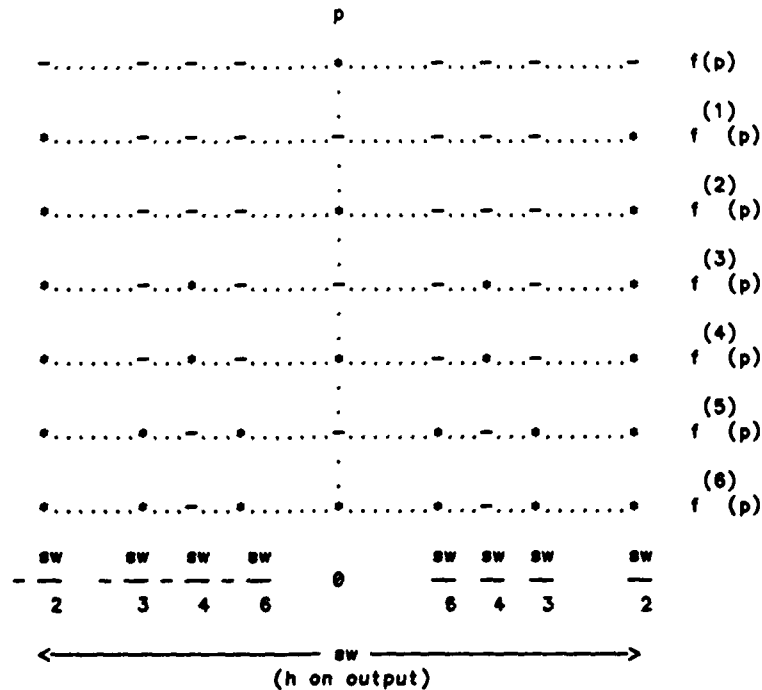
ss: (step size) the spacing between the points plotted along the line of investigation. This is  $h$  in the source code.

sw: (stencil width) the width of the interval over which the finite-difference formulas are calculated. This is  $h$  on output.

nh: the spacing between plotted points in the index space. It must be a positive integer multiple of 4.

nw: half the stencil width in index space. It must also be a positive integer multiple of 4. The number of points in index space corresponding to the stencil width is  $1 + 2 \cdot nw$ .

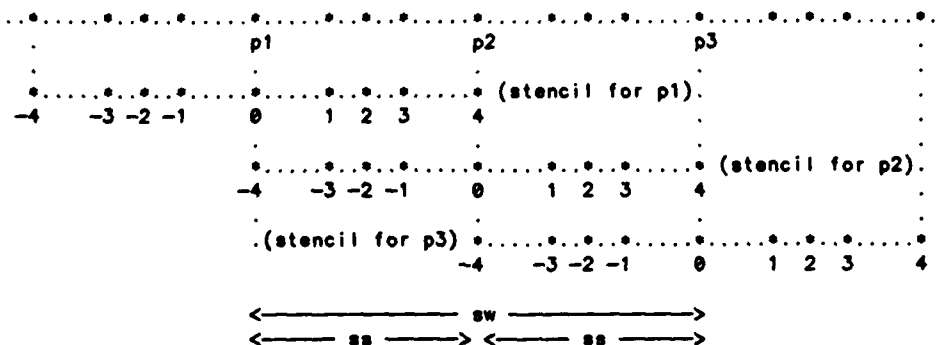
The sampling points are spaced unevenly along the line of investigation, but evenly in index space. In the following figures, asterisks denote the points at which the display function needs to be evaluated. A hyphen marks a point for which space for a function value has been provided in the arrays XS and FS, but where no evaluation is necessary. Consider a point  $p$  at which we wish to evaluate the display function and its derivatives up to order 6. The stencil is centered about  $p$ .



The figure describes the actual sampling structure along the line of investigation. Space has been provided only for points at which the display function may have to be evaluated. Providing space for sampling at increments of  $sw/12$  would be similar but wasteful since at most only 9 points (not 13) are needed in each stencil.

Next, consider the problem of extending the sampling over an entire line segment. The particular extension used in MICROSCOPE was chosen so that changing the step size and stencil width by powers of two would require minimal computation. Other factors are allowed but require greater computation.

Considering a portion of the line segment containing a sequence of three plotted points:  $p_1, p_2$ , and  $p_3$ , the spacing (for the ratio  $sw/ss = 2/1$ ,  $nw = 4$ , and assuming all tangential derivatives are to be plotted) looks like:

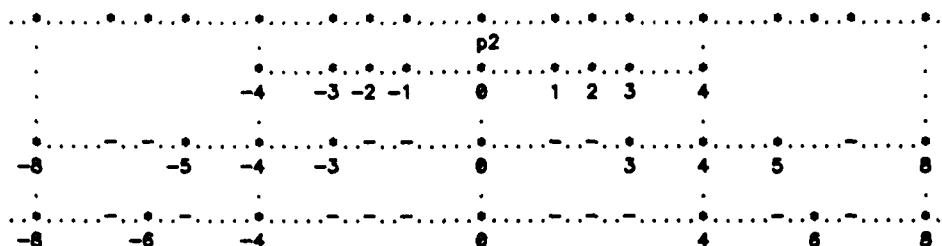


Note how the stencils overlap, allowing the use of one value of the display function for more than one (here two or three) values of a derivative. If the stencil width is doubled while the step size is kept constant (corresponding to an increase in the window width by a factor 2) we obtain:



The above operation requires additional evaluation of the display function only at the boundary of the interval of investigation. The reverse process of halving the window width would require no additional evaluations at all.

So that all derivatives, up to and including the sixth, can be evaluated, the minimal number of points between plotted points (such as p1,p2,p3) is 4. Furthermore, the distance is always a multiple of 4. Similarly, the minimal number of points needed in a stencil is 9 (i.e. -4,...,0,...,4), and it is always of the form 1+(an integer multiple of 8), although evaluation takes place at at most 9 points. These features are illustrated in the following progression of increasing stencils:



These 3 stencils span index sets of 9, 17, and 25 points, respectively (although the third set would not fit on this page). The variable *nh* for these is 4, 8, and 12, respectively.

The relation between the quantities *ss* and *ROPSTS(2)* is given by

$$ss = \frac{nh}{8} \cdot ROPSTS(2)$$

The program strives to maintain a value of *nh* = 8. For a more detailed discussion see the description of the routine *SGAMMA* below.

The relation between the point of examination, *P*, a point of evaluation, *PS*, the distance *t* of *ps* from *P*, the direction of examination, *d*, and the quantities in *MICROSCOPE*, is given by:

```

XS(ICENTR-2) = -ROPSTS(2)/4
XS(ICENTR-1) = -ROPSTS(2)/6
XS(ICENTR)   = 0
XS(ICENTR+1) = ROPSTS(2)/6
XS(ICENTR+2) = ROPSTS(2)/4
XS(ICENTR+3) = ROPSTS(2)/3
XS(ICENTR+4) = ROPSTS(2)/2
XS(ICENTR+5) = 2*ROPSTS(2)/3
XS(ICENTR+6) = 3*ROPSTS(2)/4
XS(ICENTR+7) = 5*ROPSTS(2)/6
XS(ICENTR+8) = ROPSTS(2)

```

and



```
P = ROPPNT(*,2)
d = ROPUDI(*,2)
t = XS(i)
PS = P + t * d
FS(i) = f(PS)
```

LDEF(I) = .TRUE.    <==>    FS(I) has been evaluated

To explain the calculation of the derivatives, consider a point p at which we wish to evaluate the function and its derivatives. To evaluate all derivatives up to the sixth we need 9 points. We number them as follows:

1	2	3	4	5	6	7	8	9
*.....*	*.....*	*.....*	*.....*	*.....*	*.....*	*.....*	*.....*	*.....*
.	.	.	.	p	.	.	.	.
.	.	.	.	.	.	.	.	.
sw	sw	sw	sw	0	sw	sw	sw	sw
-----	-----	-----	-----	-----	-----	-----	-----	-----
2	3	4	6		6	4	3	2

Instead of using the formulas given in section 1.2 directly, the following equivalent but more efficient formulas are used:

```
g0 = f5
g1 = f1 + f9
g2 = - ( f1 - f9 )
g3 = 4 * ( f3 + f7 )
g4 = 4 * ( f3 - f7 )
g5 = 4 * ( f2 - f8 ) + 5 * ( f6 - f4 )
g6 = - 6 * ( f2 + f8 ) + 15 * ( f4 + f6 )
```

$Q_i^{(0)} = g_0$

$Q_i^{(1)} = (1/sw) * g_2$

$$Q_i^{(2)} = (2/sw)^2 * (g_1 - 2*g_0)$$

$$Q_i^{(3)} = (4/sw)^3 / 4 * (2*g_2 + g_4)$$

$$Q_i^{(4)} = (4/sw)^4 * (g_1 - g_3 + 6*g_0)$$

$$Q_i^{(5)} = (6/sw)^5 / 2 * (g_2 + g_5)$$

$$Q_i^{(6)} = (6/sw)^6 * (g_1 + g_6 - 20*g_0)$$

The sequence in which MCRSCP calls the main routines is:

- SGAMMA - calculates gamma (see following discussion)
- CHKCMP - checks to see what changes in computations are needed
- SSMPUD - shift and copy
- SCMUPD - compute new values

When a step size/stencil width changing command is given, the program will attempt to change nh and nw without having to change ROPSTS(2) (and hence copy or throw out data). Otherwise, MCRSCP attempts to choose a change in scaling (ROPSTS(2)) which saves as much data as possible. This is the purpose of the routine SGAMMA which we now discuss in some detail. First we introduce additional notation.

- width: the number of columns on the terminal screen used for plotting
- m: scaling factor for step size (either a positive integer or 1/(positive integer) )
- v: scaling factor for stencil width (or window). (either positive integer or 1/(positive integer) )
- rho: lowest common multiple of the denominators of m and v.
- s: = ROPSTS(2). (base step size when nh = 8). (this differs from the s that denotes the interval elsewhere in this manual)

Primes will denote the new values for nh,nw,ss,and s. For example: h' denotes the step size after changes have been made.

Define:  $gamma = s/s' = num/den > 0$

Constraints: The commands set the values of  $m$  and  $v$ . There are 5 constraints which are then used to choose the value of  $\gamma$ . They are not all independent of each other.

- (1)  $4 \leq nh' \leq IXSIZE/width$
- (2)  $4 \leq nw' \leq IXSIZE/2$
- (3)  $nh'/4$  must be integer  $\iff mod(nh',4) = 0$
- (4)  $nw'/4$  must be integer  $\iff mod(nw',4) = 0$
- (5) effective sampling width =  $nh' \cdot width + 2 \cdot nw' \leq IXSIZE$   
in the index space

We have:

$$ss = (nh/8) \cdot s \quad \text{and} \quad ss' = (nh'/8) \cdot s'$$

Also:

$$ss' = m \cdot h$$

Hence:

$$ss' = (nh'/8) \cdot s' = m \cdot h = m \cdot (nh/8) \cdot s$$

$$\implies (nh'/8) \cdot s' = m \cdot (nh/8) \cdot s$$

$$\implies nh' = m \cdot nh \cdot s/s' = m \cdot nh \cdot \gamma$$

Similarly:

$$sw = (nw/4) \cdot s, \quad sw' = (nw'/4) \cdot s', \quad \text{and} \quad sw' = v \cdot sw$$

$$\implies sw' = (nw'/4) \cdot s' = v \cdot sw = v \cdot (nw/4) \cdot s$$

$$\implies nw' = v \cdot nw \cdot s/s' = v \cdot nw \cdot \gamma$$

In summary:

$$nh' = m \cdot nh \cdot \gamma$$

$$nw' = v \cdot nw \cdot \gamma$$

Referring to the conditions numbered above, divide (1) by  $m \cdot nh$  and (2) by  $v \cdot nw$  to get:

$$\frac{4}{m \cdot nh} \leq \gamma \leq \frac{IXSIZE}{width} \cdot \frac{1}{m \cdot nh}$$

and

$$\frac{4}{v * nw} \leq \text{gamma} \leq \frac{\text{IXSIZE}}{2} * \frac{1}{v * nw}$$

Let

$$a = \max\left(\frac{4}{m * nh}, \frac{4}{v * nw}\right)$$

and

$$b = \min\left(\frac{\text{IXSIZE}}{\text{width}} * \frac{1}{m * nh}, \frac{\text{IXSIZE}}{2} * \frac{1}{v * nw}\right)$$

From above, gamma must satisfy:

$$a \leq \text{gamma} \leq b$$

Furthermore:

$$nh' = m * nh * \text{gamma} \quad \text{and} \quad \text{mod}(nh', 4) = 0$$

$$\Rightarrow (*) \quad \text{mod}(m * nh * \text{gamma}, 4) = 0$$

$$nw' = v * nw * \text{gamma} \quad \text{and} \quad \text{mod}(v * nw * \text{gamma}, 4) = 0$$

$$\Rightarrow (**) \quad \text{mod}(v * nw * \text{gamma}, 4) = 0$$

The routine SGAMMA uses the above two conditions (\*) and (\*\*) to choose acceptable values for gamma (and hence num and den since gamma = num/den).

### Outline of SGAMMA

- check to see if the changes m and v can be accommodated by changing nh and nw, but not ss (h in code).
- if no changes in ss are needed
- ← then num = den = 1 ==> gamma = 1 and return
- calculate a and b
- if m = v
- ← then set gamma = m, ROPSTS(2) = m \* ROPSTS(2), and return
- calculate rho
  - $m' = \rho * m$  ( $m'$  and  $v'$  are positive integers)
  - $v' = \rho * v$
- determine "all" possible numerators and denominators for gamma (up to 32) which satisfy:
  - $a \leq \text{num/den} \leq b$
  - $\text{mod}(m * nh * \text{gamma}, 4) = 0$
  - $\text{mod}(v * nw * \text{gamma}, 4) = 0$
- rank the candidate values of gamma according to 4 criteria
  - (1)  $nh' * \text{width} + 2 * nw' \leq \text{IXSIZE}$
  - (2)  $8 * \text{width} \leq nh' * \text{width} + 2 * nw' \leq \text{ICENTR}$   
(this allows for halving and doubling without rescaling)
  - (3) gamma a power of 2 (positive or negative)
  - (4)  $nh' > 4$  and  $nw' > 4$
- choose the combination of num and den which has the highest corresponding rank and set the variables accordingly.
- ←• return

### II.3 Common Blocks

Here we list the COMMON blocks used in MICROSCOPE. This information may be useful in modifying the code and in accessing the data structures of the package via the command USER (see section 4.15). For each common block we list in parentheses the number of words (i.e. 2 words for each DOUBLE precision variable and 1 word for each REAL, INTEGER, or LOGICAL variable) occupied by the block. The blocks are listed in the same sequence as they appear in the routine MCRSCP.

CB (16) controls cross differentiation.

```

CBDSW      : width of differentiation stencil
CBD(3)     : normalized cross direction
CBDU(3)    : unnormalized cross direction
ICBD       : order of cross differentiation
LCBD       : .TRUE. if cross differentiation is on, .FALSE.
              otherwise
  
```

IQ (5) contains the input/output device numbers.

RECORD : Record device number. (Output data file for recording  
screen images and comments)  
GRAPHD : Graphics device number. (terminal or graphics display)  
HELPD : Help file device number.  
RSTRTD : Restart file device number. (unformatted output data  
file for use in restarting at a later date)  
INPUTD : Input device number. (usually the terminal)

LOG (1) controls logging

LCHN : Logging device number (= 0  $\longleftrightarrow$  logging is off)

LOGCOM (5391) contains logical arrays used primarily by SCMUPD  
to keep track of what has already been computed.

LDF(7) : Denotes whether the (i-1)'th derivative has been calcu-  
lated. LDF(i) = .FALSE. means the (i-1)'th derivative  
has not been calculated. LDF(i) = .TRUE. means it has.  
LPLT(7) : Denotes whether the (i-1)'th derivative is currently  
being displayed. LPLT(i) = .FALSE. means the (i-1)'th  
derivative is not being displayed and LPLT(i) = .TRUE.  
means it is.  
LDEF(5377) : Denotes whether or not the interpolation function has  
been calculated at the coordinates corresponding to  
XS(i). (i.e. whether FS(i) has been assigned)  
LDEF(i) = .FALSE. means the i'th value has not been  
sampled and LDEF(i) = .TRUE. means it has.

FUNCOM (23,426) contains arrays pertaining to the values of the  
interpolation function at its sampled points.

XS(5377) : The distance array, always centered about 0. Its defini-  
tion and usage is given in the second part of this  
appendix.  
FS(5377) : The array containing the values of the interpolation  
function corresponding to the points specified by the  
XS() array. See the second part of this appendix for  
discussion.  
DF(135,7) : The finite differences. DF(i,j) corresponds to the i'th  
plotted point and (j-1)'th derivative. These are the  
p1,p2,p3's of the discussion in the second part of  
this appendix.  
DFMINMX(2,7): These are the minimums/maximums of the finite differences  
in the DF() array.

ERRCOM (2) used to handle error checking and reporting.

When errors are detected in any of the sub routines or computation routines, the errors are flagged and then reported by the CHKERR routine.

ERRCOD : This is an integer code number for the type of error detected.  
ERR : This is the logical flag which denotes whether or not an error has occurred. ERR = .FALSE. means no error has occurred and ERR = .TRUE. means an error has occurred.

PLTCOM (16342) contains display versions of the derivatives.

IPLLOT(135,7): IPLLOT(i,j) is the "screen coordinate" (integer) of the (j-1)'th derivative in the i'th column.  
ISCRN1(135,57): The previous screen image. ISCRN1(i,j) contains the Hollerith representation of the character to be displayed at the (i,j)'th pixel on the screen. (where indexing starts at the upper left corner of the screen). i is the horizontal component and j is the vertical component.  
ISCRN2(135,57): The current screen image. This has the same description as ISCRN1(.) above.  
SCALE(7) : The scale factors used to individually scale, up or down, the finite difference "curves" (in the DF(.) array) so that the curve fills up the entire screen in the sense that the maximum point lies on the top row and minimum point lies on the bottom row of the screen.

LOADIO (3) contains channel numbers of the loading devices.

PLOAD : point loading device  
DLOAD : direction loading device  
CLOAD : cross direction loading device

SCREEN (7) contains the values of parameters describing the device on which most interaction occurs (i.e. the screen of the terminal in most cases)

OUTPUT	:	The device number of the graphics device (usually the terminal)
LINES	:	The number of usable lines on the screen. Note that this may have to be one less than the number of displayable lines.
WIDTH	:	The number of columns to be displayed. It is best to choose an odd number for this so that the center can be plotted. It must be less than or equal to 135. 75 is a common value.
ILP	:	The number of lines in the curve plotting portion of the display. It must be less than or equal to 57.
IPRMPT	:	The number of lines above the screen bottom at which the prompting commands are given. It must be at least 2.
IDSPLA	:	The number of lines above the screen bottom at which the numerical data region begins. It must be greater than IPRMPT by 6 or more.
LSCRN	:	The logical variable denoting whether the curves are currently being displayed or they have been overwritten by text. This is to allow the program to take advantage of terminals which allow screen editing. LSCRN = .FALSE. means that the screen has been overwritten since the previous command and LSCRN = .TRUE. means that the screen is intact from the previous command.

OPTION (122) contains most of the variables, other than the sampling arrays in FUNCOM, which may change due to commands. There are 2 copies of each variable. (1) is the previous value and (2) is the current value. If mistakes are made in the inactive (deactiv) mode (i.e. LGO = .FALSE.) then the UNDO command will erase the changes and copy the old set (1) to (2).

ROPDI(3,2)	:	The vector specifying the direction of investigation. Note that for all vector valued quantities, the dimension is specified by IOPDM() below.
ROPPNT(3,2):		The vector specifying the center of the line of investigation. If the shift is 0 (i.e. IOPSH(2) = 0) then this is the point which is at the center of the screen.
ROPDR1(3,2):		Currently unused vector.
ROPDR2(3,2):		Currently unused vector.
ROPSTS(2)	:	The nominal spacing between the points at which the finite differences are calculated. These will be the plotted points. "Nominal" refers to the base step size when nh(2) = 8, as discussed in the second



part of this appendix. As noted earlier, this is not the h which appears on output.

ROPSTW(2) : The stencil width (sw). This corresponds to the h which appears on output. It is discussed in the second part of this appendix.

ROPUDI(3,2): The normalized (unit) vector specifying the direction of investigation. This is the normalized version of the ROPDI() vector.

NH(2) : The step size in index space. See the discussion in the second part of this appendix.

NW(2) : Half of the stencil width in index space. See the discussion in the second part of this appendix.

ILEFT(2) : The index of the left-most element of the FS() array which has been assigned a value.

IRIGHT(2) : The index of the right-most element of the FS() array which has been assigned a value.

IOPDM(2) : The dimension of the domain space. It can be 1, 2 or 3.

IOPSH(2) : The shift, in integer increments, of the plotted points from the center defined by ROPSTS(\*,2).

IOPSS(2) : The scaling factor for the step size. It is set by the commands.

IOPWN(2) : The scaling factor for the stencil width. This multiplies/divides the current window, as it appears on the screen. It is set by the commands.

OPDC(2) : The logical flag determining whether or not the center of the screen is to be marked.

OPDS(2) : The logical flag determining whether or not a scale is to be displayed along the horizontal axis.

OPDX(2) : The logical flag determining whether or not a line of dashes will be displayed to represent the x-axis.

OPCMP(2) : The logical flag denoting whether or not any computations have yet been done for the current set of ROPNT and ROPDI.

OPSPL(2) : Currently not in use.

OPDF1(7,2) : The logical array specifying which derivatives are to be displayed. It is set by the DGRAPH and EGRAPH commands.

OPDF2(7,2) : The logical array specifying which derivatives are to be accentuated as they are displayed. It is set by the ACCENT command.

NRMLZE (1) controls normalization of derivatives

NORMAL : = .TRUE. <==> normalization is on

**ROOM** (2) controls whether one line of the graphical display is transferred to the numerical display. This is only necessary if all six tangential derivatives of a cross derivative are plotted.

ILPUSR : User specified space for the graphical display  
IDSUSR : User specified space for the numerical display  
The corresponding variables ILP and IDSPLA in the common block SCREEN are modified as necessary.

**HELPER** (7330) contains help information

HELP : Help device channel number  
JHELP1 : Length of the HSUMRY information (in lines)  
JHELP2 : Length of the HELP information (in lines)  
JHELP3 : Number of available commands  
JHELP(99,2): (1,1) line in which detailed help for I-th command begins  
(1,2) length of detailed help  
IHELP(72,99): (\*,1) prompt text for I-th command

**USER** (5) see appendix III

**PLOWN** (296) contains current PLOT options

ITITLE(72) : Title  
IBOTTM(72) : Legend  
NUMBR : page number  
BL(20) : left bottom label  
BR(20) : right bottom label  
FRAME : = .TRUE. <==> frame is drawn  
COLOR : = .TRUE. <==> plot is in color  
NUMRCL : = .TRUE. <==> numerical display is drawn  
LMARK : = .TRUE. <==> graphs are marked  
LBLS : = .TRUE. <==> graphs are labeled  
DATE : = .TRUE. <==> date is drawn  
TIME : = .TRUE. <==> time is drawn

**EQOWN** (1) controls reevaluation after the command FORCE has been issued.

LFO : = .TRUE. <==> reevaluate

### Appendix III: The Test Package

The examples in this manual can be reproduced, and experience with MICROSCOPE can be gained, by using the test routines (a special choice of the trial function F and the user intervention routine SUBUSR, plus a few support routines) supplied with the package. The test code is contained in the files USRLC and USRUC (see chapter 5).

Several different trial functions are available for examination, and rounding to any number of digits up to the number supported on your local installation can be simulated.

All options are selected using the command USER, which puts you into user mode. You can change options similarly as in the PLOT mode until you are satisfied, and then return to the command mode.

Options are chosen by passing numbers to the program according to the prompts given. A current list of options is maintained on the screen.

The trial function is selected by giving its reference number. The parameter eta is 1 by default and can be changed by typing -3. By default rounding to 10 digits is simulated. Rounding can be changed by typing -1 and giving the desired number of significant digits. It can be turned off by giving 0 as the number of digits. The following trial functions are available:

- 1:  $F(x) = s(0, \text{eta}, x)$  or  $F(x) = s(0, \text{eta}, x) + e^x$
- 2:  $F(x) = s(1, \text{eta}, x)$  or  $F(x) = s(1, \text{eta}, x) + e^x$
- 3:  $F(x) = s(2, \text{eta}, x)$  or  $F(x) = s(2, \text{eta}, x) + e^x$
- 4:  $F(x) = s(3, \text{eta}, x)$  or  $F(x) = s(3, \text{eta}, x) + e^x$
- 5:  $F(x) = s(4, \text{eta}, x)$  or  $F(x) = s(4, \text{eta}, x) + e^x$
- 6:  $F(x) = s(5, \text{eta}, x)$  or  $F(x) = s(5, \text{eta}, x) + e^x$

The addition of the exponential in the above functions is off by default, and is toggled on or off by typing -2.

- 7:  $F(x) = e^{\text{eta} \cdot x}$

- 8:  $F(x,y) = \text{eta} \cdot \text{abs}(x) \cdot x^2 + (1-\text{eta}) \cdot \text{abs}(y) \cdot y$
- 9:  $F((x,y) = x^2 \cdot \text{abs}(x) \cdot y \cdot \text{abs}(y)$
- 10:  $F(x) = \text{the spline interpolant to } e^x \text{ used in section 2}$
- 11:  $F(x) = e^x - \text{function 10}$   
(i.e.  $F(x)$  is the error in the spline 10)
- 12:  $F(x) = 0$
- 13:  $F(x) = \text{eta} \cdot x$
- 14:  $F(x) = x^{\text{eta}}$
- 15:  $F(x) = x^{\text{eta}}$

The user mode is left by typing zero (or just RETURN).

The commands STORE and RESTART are effective on the test parameters which are stored in the COMMON block USER. The parameters have the following meanings:

ETA: DOUBLE PRECISION variable containing the value of eta.

IRound: Number of digits to which function values are rounded (IRound = 0  
←→ no rounding is taking place).

N: reference number of the trial function.

ADD: LOGICAL variable that indicates whether an exponential term is to be added to the trial functions defined by  $N = 1, \dots, 6$ .

The following table gives the USER defaults and the parameter settings necessary to generate the figures in this manual. N is the reference number of the trial function, D is the number of digits rounded to. The entry "—" means "does not apply". The default settings are effective on the first USER and after giving the command SETDF. Other parameters necessary for reproducing the examples in this manual (point, direction, dimension, interval etc.) can be read off the alphanumerical display in each figure.

Figure	N	D	ETA	exponential added
Default	1	10	1	NO
1	9	10	—	—
2-6	11	10	—	—
7-8	10	10	—	—
9	12	10	—	—
10	13	10	2D-9	—
11-12	10	10	—	—
13-18	3	10	-5D-3	YES
19	1	10	-1	NO
20	2	10	-1	NO
21	3	10	-1	NO
22	4	10	-1	NO
23	5	10	-1	NO
24	6	10	-1	NO
25-26	7	10	1	—
27	9	10	—	—
28	14	10	1.5	—
29	15	10	1.5	—

Table of test parameters

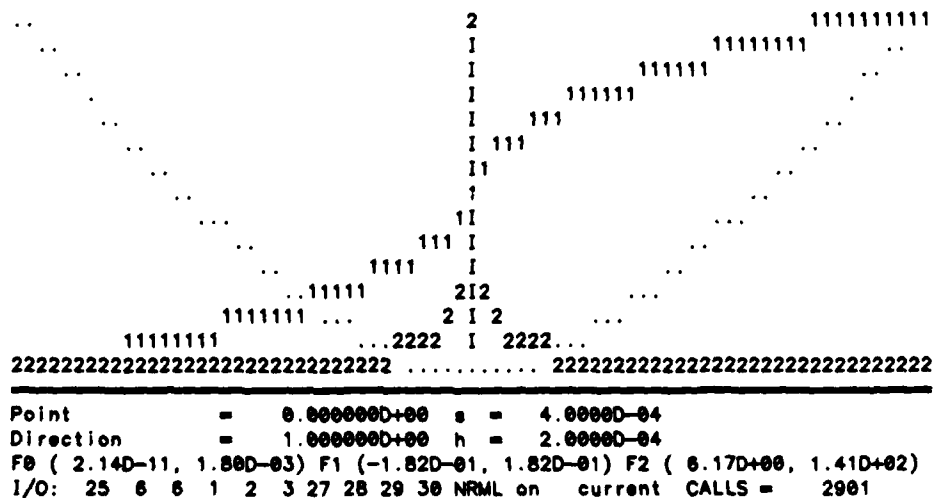
Because of the random nature of round-off errors, somewhat different displays may be obtained even with the above parameter settings.

Some of the above test functions so far have not been used in this manual. They illustrate more subtle phenomena that will not normally occur in multivariate interpolation and approximation. In a typical application, the trial function will have a finite degree of differentiability across the front, and be infinitely often differentiable in any direction contained in the front. Function 8 illustrates a situation where the degree of differentiability is finite in any direction, but is dependent upon the direction of investigation. The function is once differentiable in the y direction and twice differentiable in the x direction. The relative sizes of the jumps in the appropriate derivatives can be controlled by the parameter eta.

The functions 14 and 15 can be used to illustrate discontinuities other than step discontinuities. The functions differ in the way they are defined for negative values of x, for which the value of function 14 is negative, and that of function 15 positive, if the integer part of abs(eta) is even, and of the opposite sign if it is odd.

For both functions 14 and 15, the value at  $x = 0$  is 0 if eta is negative. This prevents a floating point overflow and a meaningless display.

The following two figures illustrate functions 14 and 15 for  $\epsilon = 1.5$  (i.e. a function that is 1.5 times differentiable). The first derivative is continuous, but the second derivative exhibits a pole. The illustration of these features is enhanced by choosing the window width to be less than 1.



**Figure 28 A  $3/2$  times differentiable function**

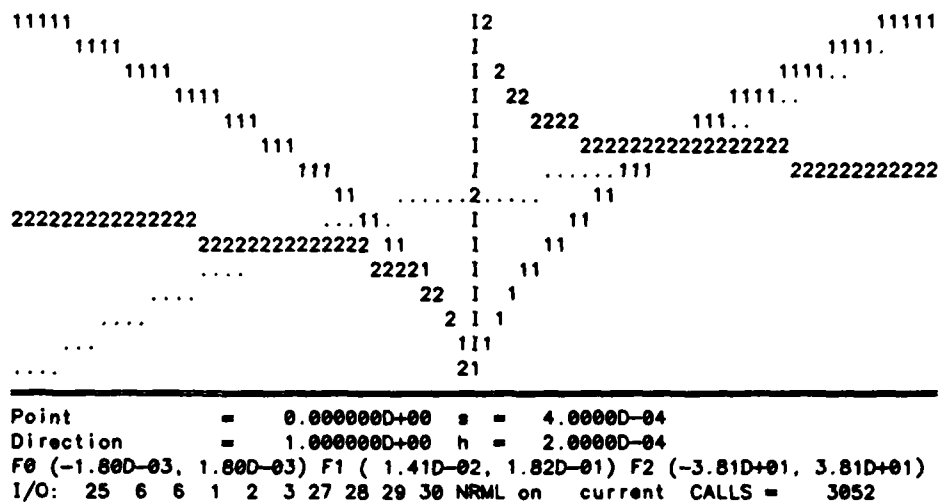


Figure 29 Another 3/2 times differentiable function

Figure 29 shows a cusp in the first derivative whereas in figure 28 the first derivative passes vertically through the origin. Both derivatives are continuous at the point of examination and have a vertical tangent there. The fact that the second derivative has a pole can be verified by observing that the extreme function value increases as the discretization parameter is decreased. For a cusp, the extreme function value approaches a limit as  $h$  decreases. This phenomenon can be utilized to distinguish between a cusp and a pole.

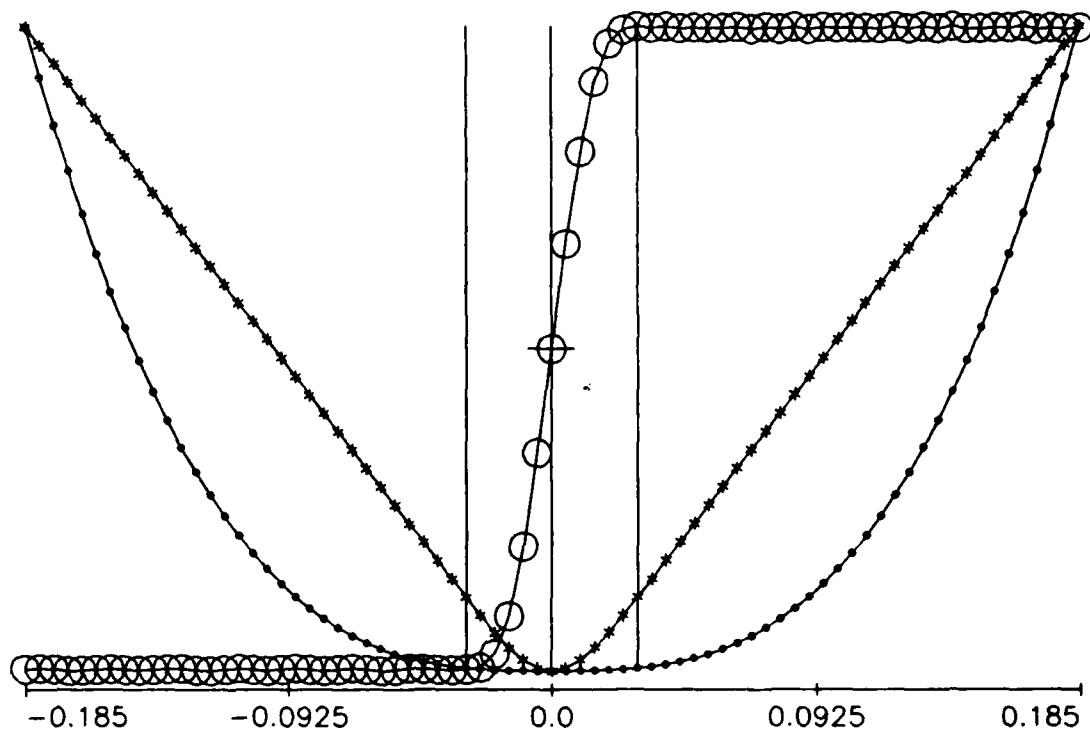
It is of course straightforward to program other or additional examples for a trial function by modifying the routines in the test package appropriately.

Appendix IV: Examples for Graphical (<PLOT79>) Displays

The following pages contain <PLOT79> Equivalents of the figures in this  
manual



## MICROSCOPE: A Software System for Multivariate Analysis



CD: 1 Dir = (0.0, 1.000)

ch = 6.00E-04

Point = (0.0, 1.000)

s = 5.00E-03

Direction = (1.000, 0.0)

h = 0.0300

F0 \* (9.386E-09, 0.01266)

F2 \* (0.1200, 2.220)

F3 O (-12.04, 12.07)

Number of Evaluations: 174

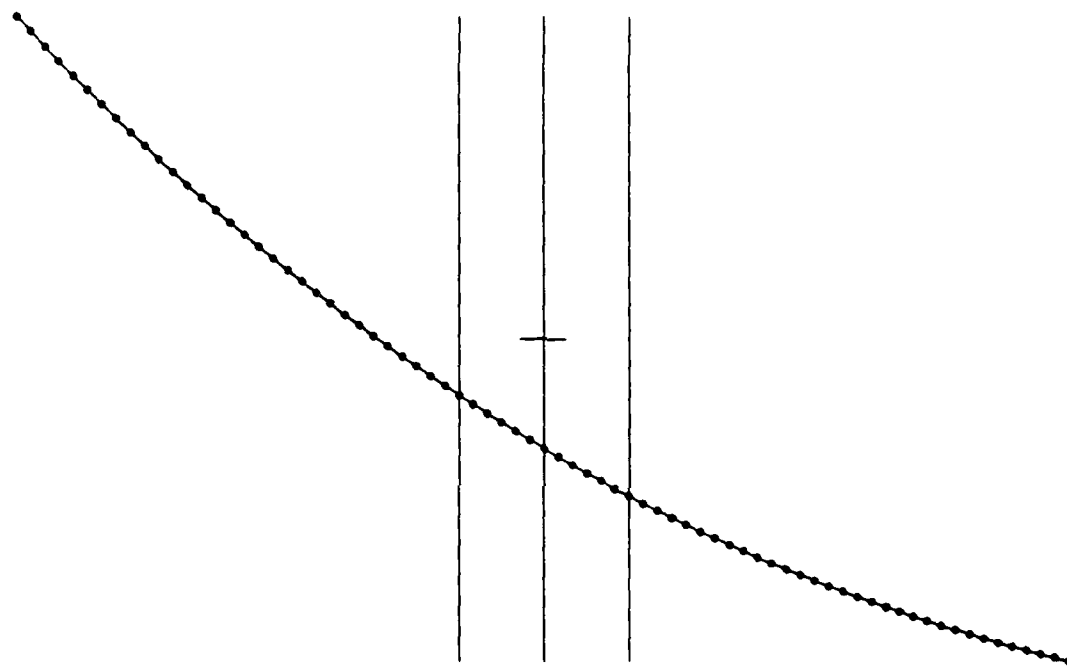
This plot corresponding to figure

1

09:21:41

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

FO • (-0.03332, 0.06764)

s = 5.00E-03

h = 0.0300

Number of Evaluations: 249

This plot corresponding to figure

2

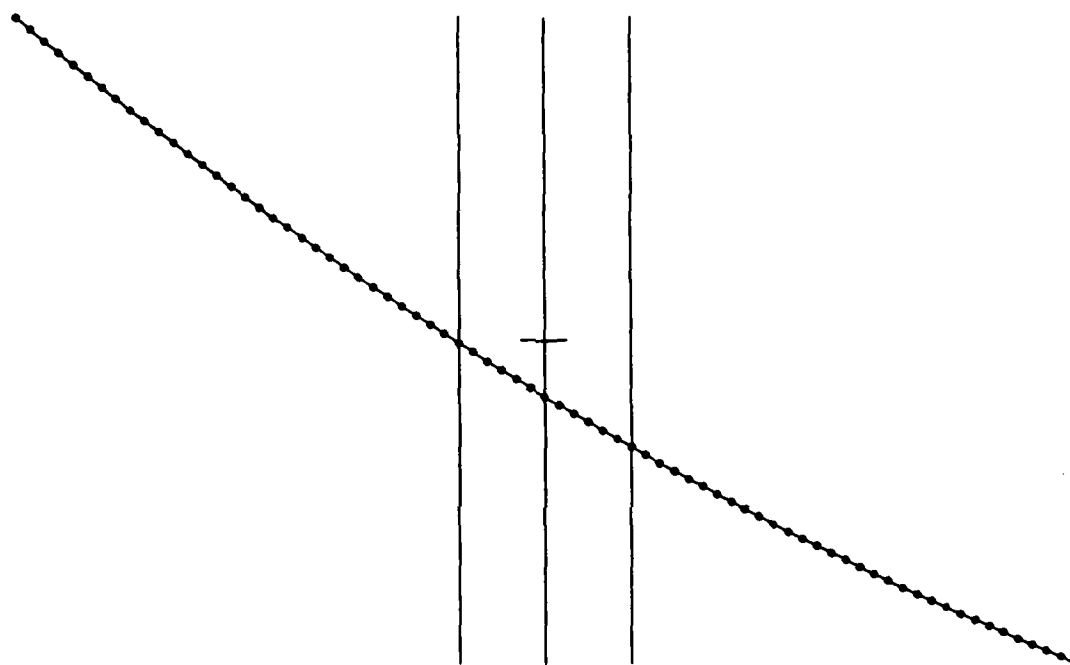
P L O T 7 9

University of Utah

09:22:07

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F0 • (-0.02027, 0.02884)

s = 2.50E-03

h = 0.0150

Number of Evaluations: 287

This plot corresponding to figure

3

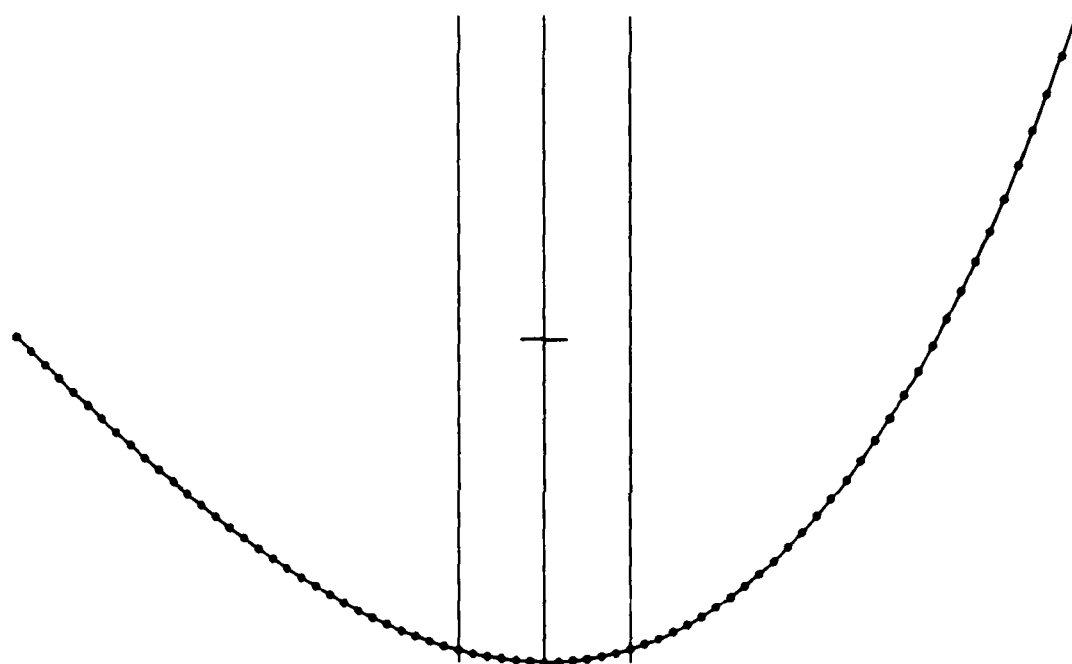
P L O T 7 9

University of Utah

09:22:50

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 2.00

Direction = 1.000

F0 • (2.638E-11, 0.01415)

s = 5.00E-03

h = 0.0300

Number of Evaluations: 362

This plot corresponding to figure

4

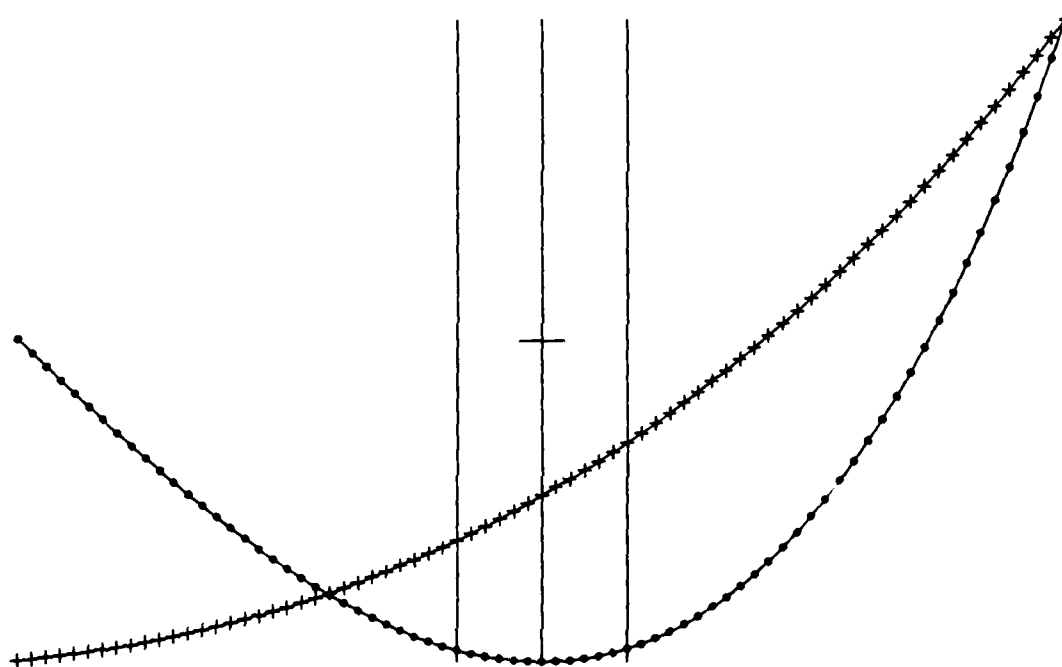
P L O T 7 9

University of Utah

09:23:37

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 2.00

Direction = 1.000

F0 = (2.638E-11, 0.01415)

Number of Evaluations: 374

s = 5.00E-03

h = 0.0300

F1 = (-0.06111, 0.1768)

This plot corresponding to figure

5

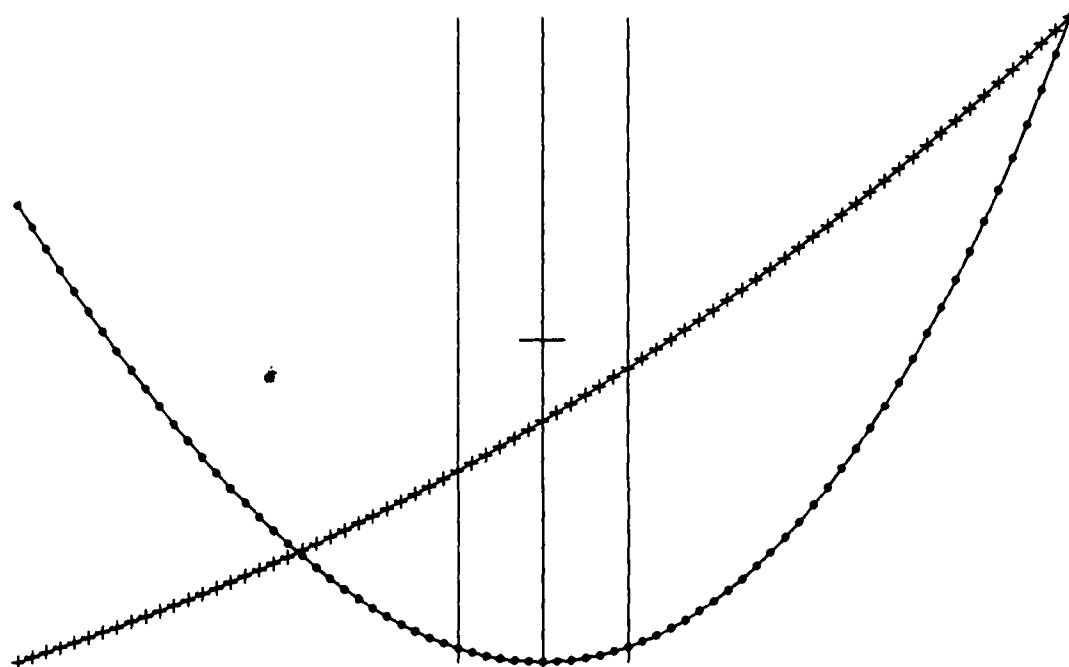
PLOT 79

University of Utah

09:24:01

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 2.00

Direction = 1.000

F0 = (2.638E-11, 3.029E-03)

Number of Evaluations: 418

s = 2.50E-03

h = 0.0150

F1 = (-0.04209, 0.07089)

This plot corresponding to figure

6

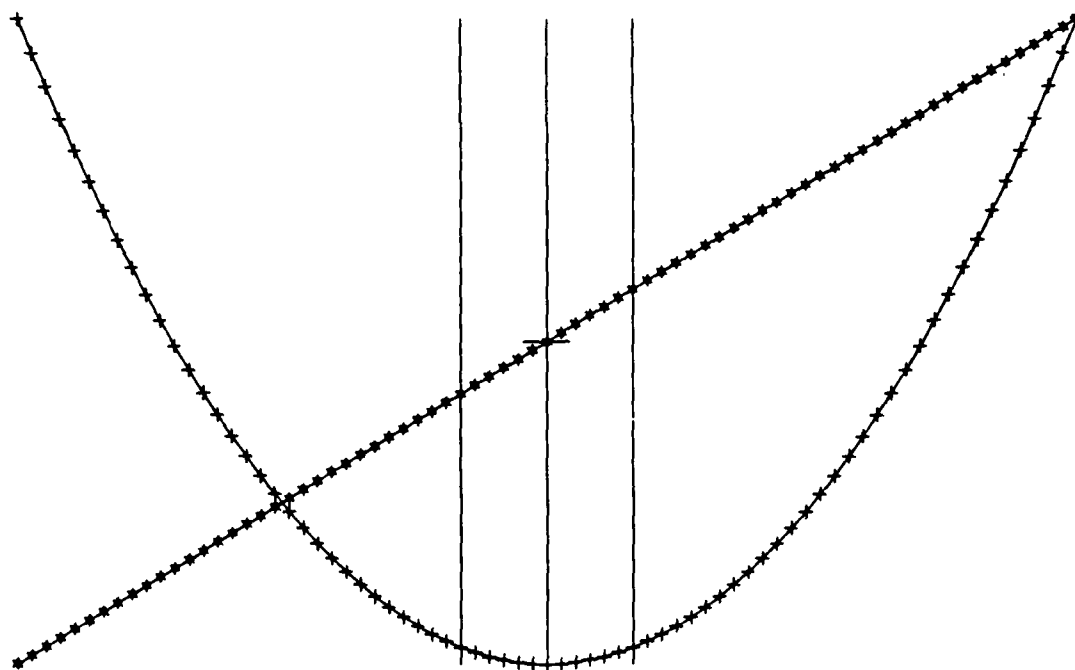
P L O T 7 9

University of Utah

09:24:30

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F0 • (0.8828, 1.117)

F2 \* (-0.2526, 0.2526)

s = 2.50E-03

h = 0.0150

F1 + (1.263, 1.274)

Number of Evaluations: 505

This plot corresponding to figure

7

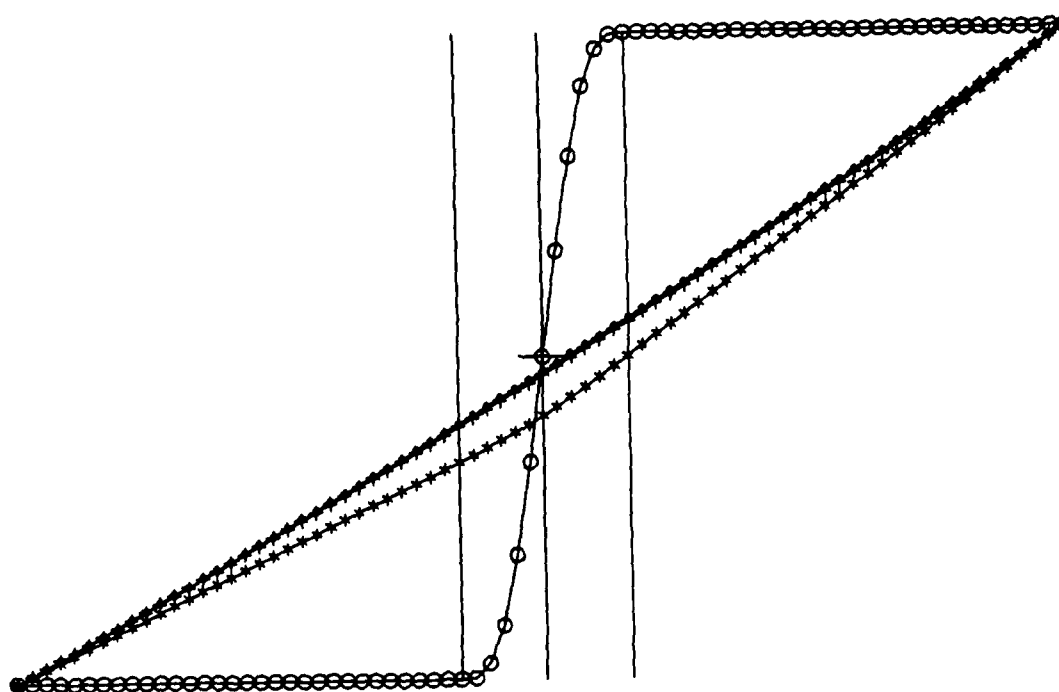
PLOT 79

University of Utah

09:25:10

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 1.000

Direction = 1.000

F0 • (2.486, 2.973)

F2 \* (2.478, 3.106)

Number of Evaluations: 592

s = 2.50E-03

h = 0.0150

F1 + (2.387, 2.898)

F3 o (2.730, 4.058)

This plot corresponding to figure

8

PLOT 79

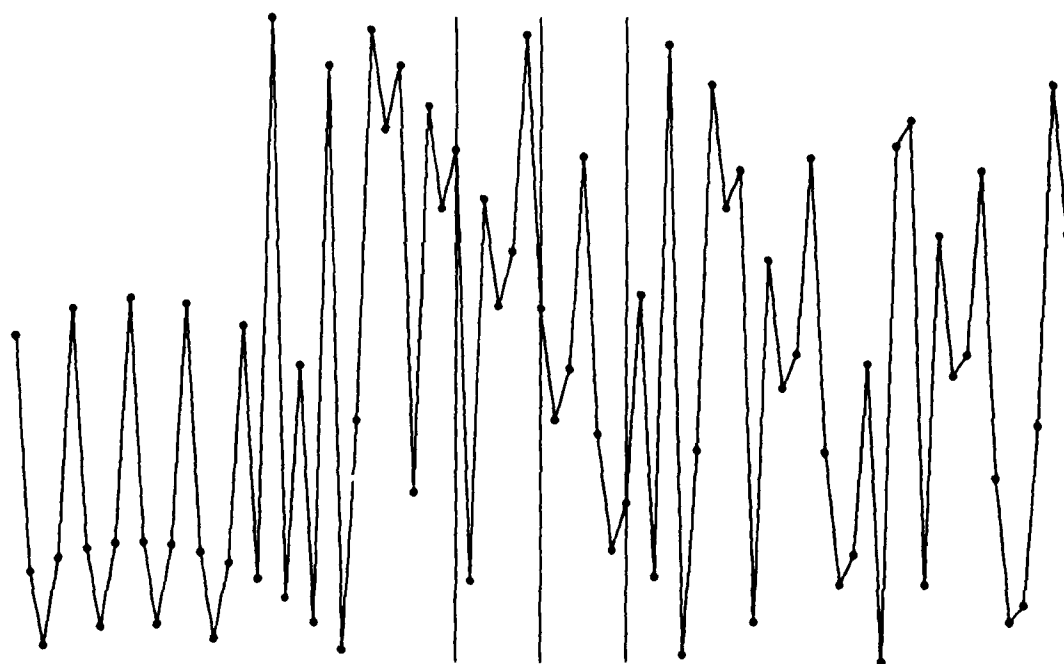
University of Utah



09:25:36

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.234

Direction = 1.000

F0 = (-9.993E-11, 9.780E-11)

s = 2.50E-03

h = 0.0150

Number of Evaluations: 667

This plot corresponding to figure

9

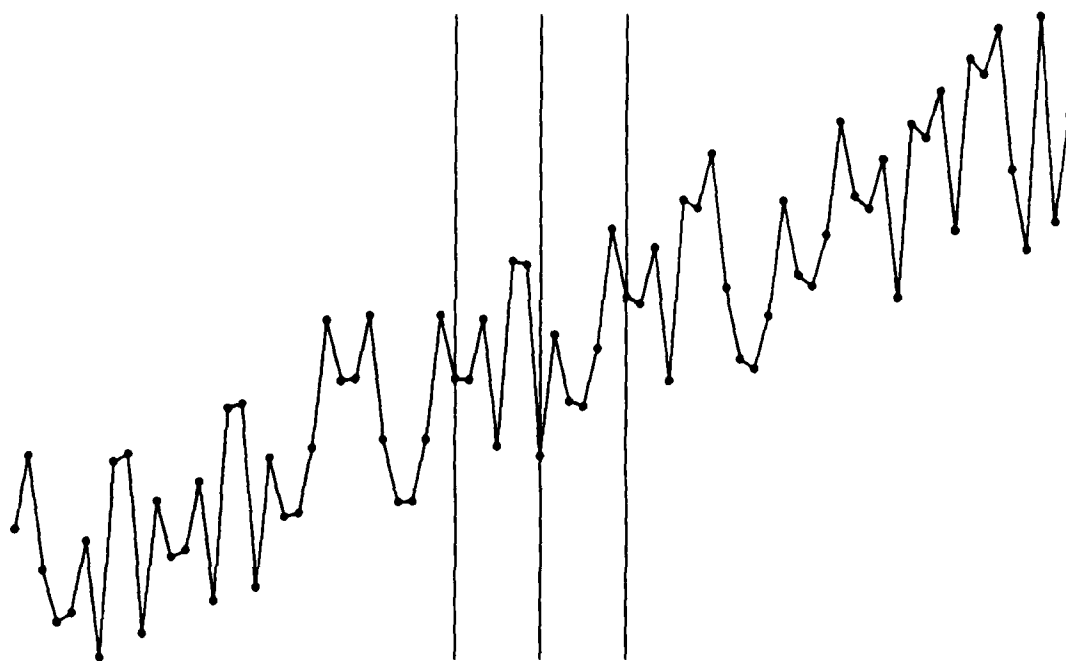
PLOT 79

University of Utah

09:26:32

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.234

Direction = 1.000

F0 = (2.162E-10, 7.314E-10)

s = 2.50E-03

h = 0.0150

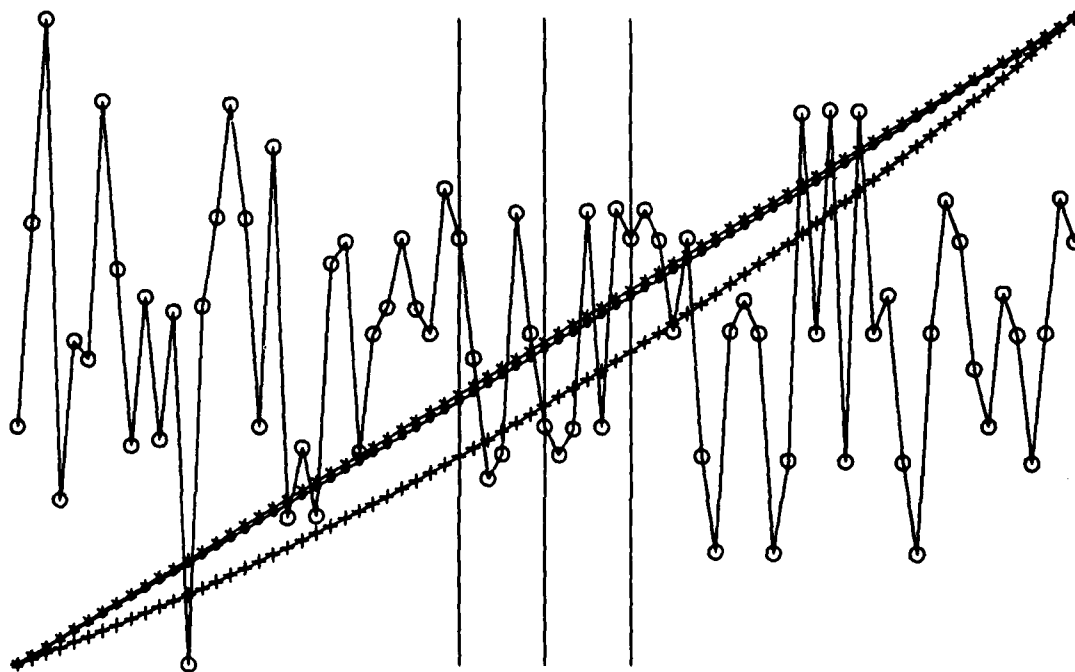
Number of Evaluations: 742

This plot corresponding to figure  
10

PLOT 79

University of Utah

## MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.234

Direction = 1.000

F0 • (1.180, 1.428)

F2 \* (0.3871, 0.8924)

Number of Evaluations: 841

s = 2.50E-03

h = 0.0150

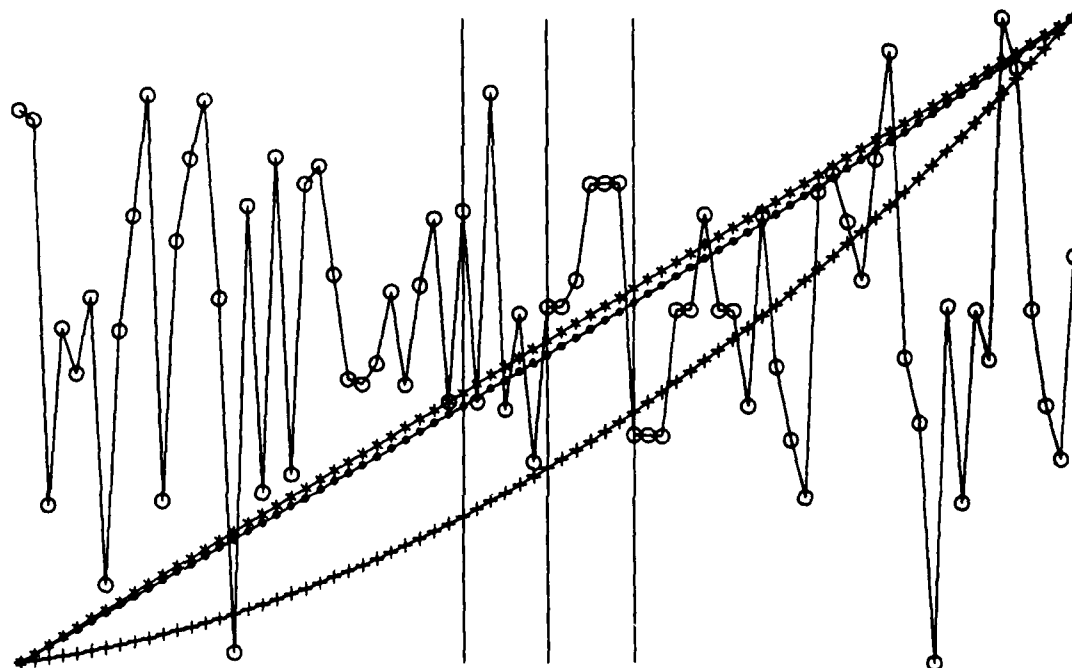
F1 + (1.290, 1.408)

F3 o (2.730, 2.732)

This plot corresponding to figure

11

## MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.234

Direction = 1.000

F0 • (1.062, 1.563)

F2 \* (0.1344, 1.145)

Number of Evaluations: 885

s = 5.00E-03

h = 0.0300

F1 + (1.266, 1.503)

F3 o (2.731, 2.731)

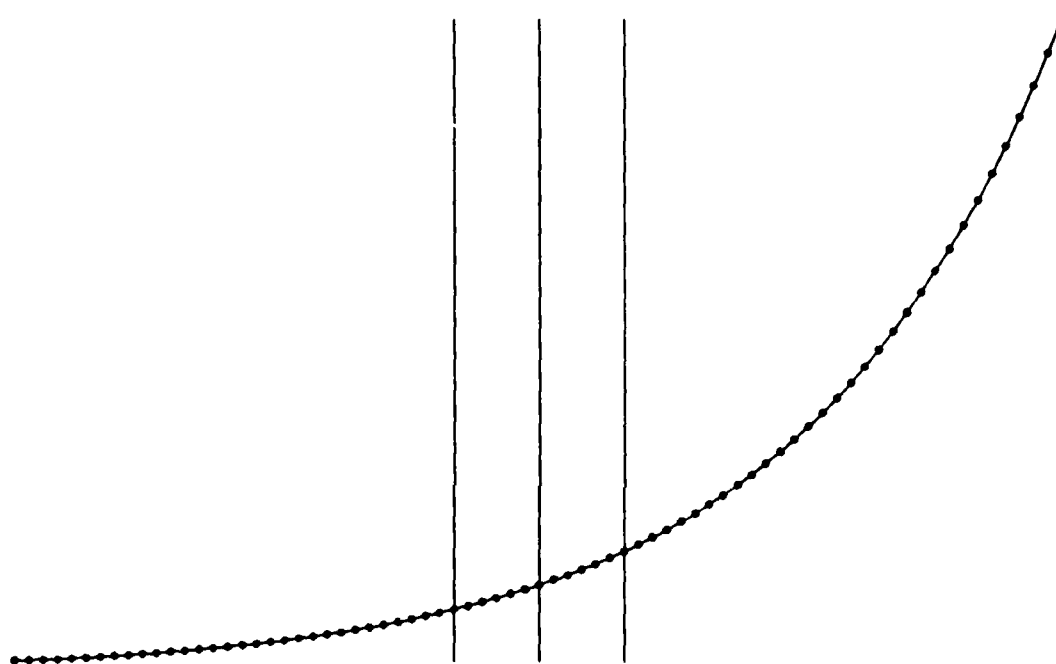
This plot corresponding to figure

12

09:32:16

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 1.000  
Direction = 4.00  
FO • (0.3678, 20.06)

s = 0.0540  
h = 0.324

Number of Evaluations: 960

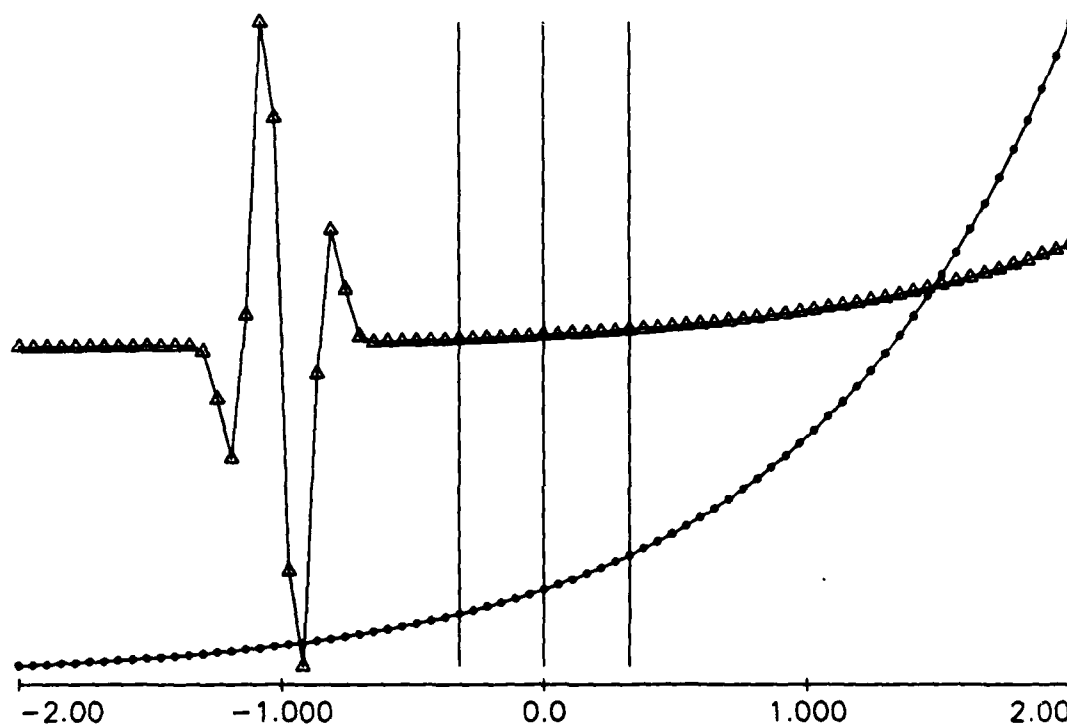
This plot corresponding to figure

13

PLOT 79

University of Utah

## MICROSCOPE: A Software System for Multivariate Analysis



Point = 1.000  
Direction = 4.00  
FO = (0.3678, 20.06)  
Number of Evaluations: 972

s = 0.0540  
h = 0.324  
F6 = (-60.68, 62.69)

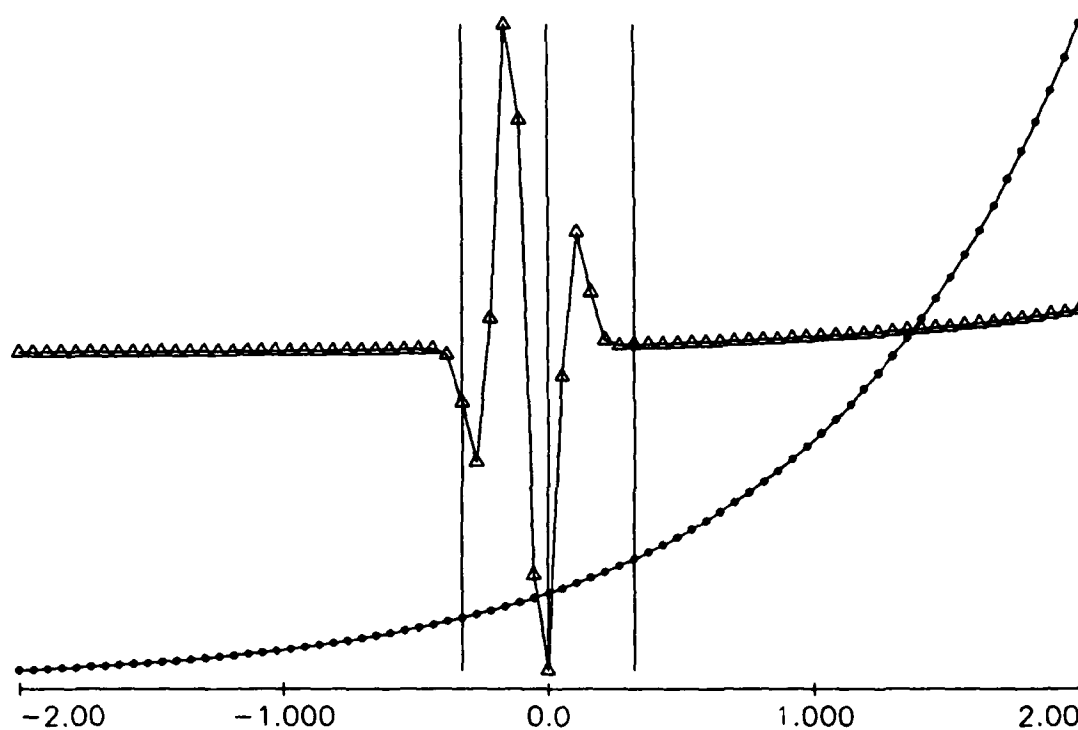
This plot corresponding to figure

14

09:33:54

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 1.000

Direction = 4.00

F0 • (0.1467, 8.002)

Number of Evaluations: 989

s = 0.0540

h = 0.324

F6 Δ (-60.68, 62.69)

This plot corresponding to figure

15

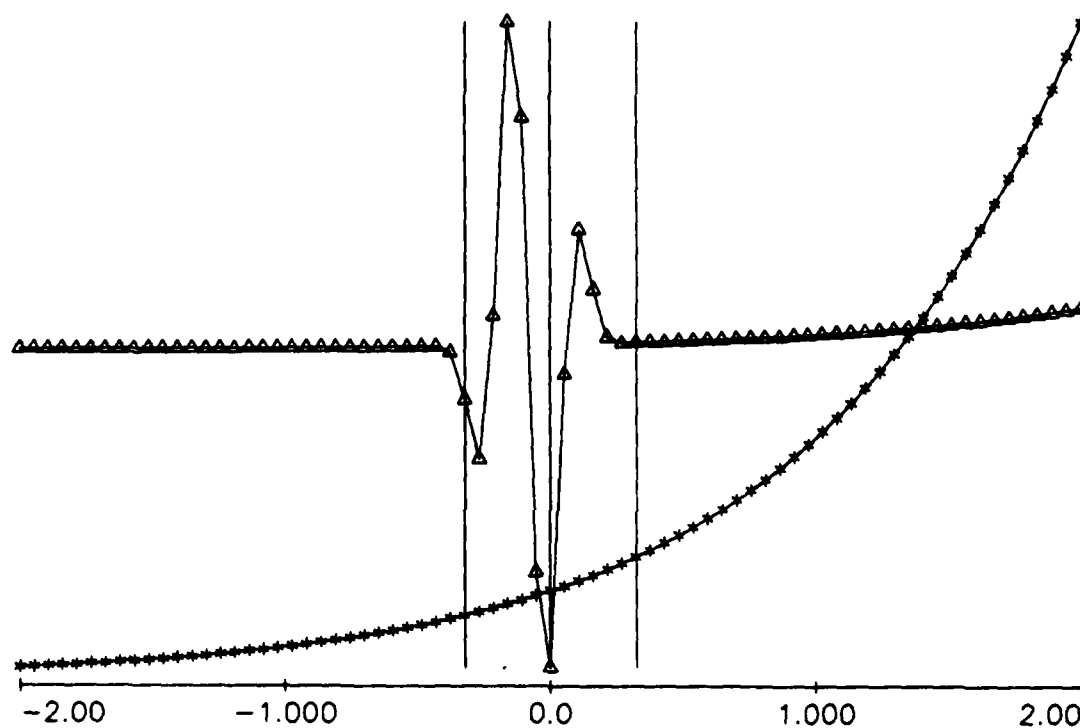
PLOT 79

University of Utah

09:34:27

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 1.000

Direction = 4.00

F2 \* (0.1480, 8.078)

Number of Evaluations: 989

s = 0.0540

h = 0.324

F6  $\Delta$  (-60.68, 62.69)

This plot corresponding to figure

16

P L O T 7 9

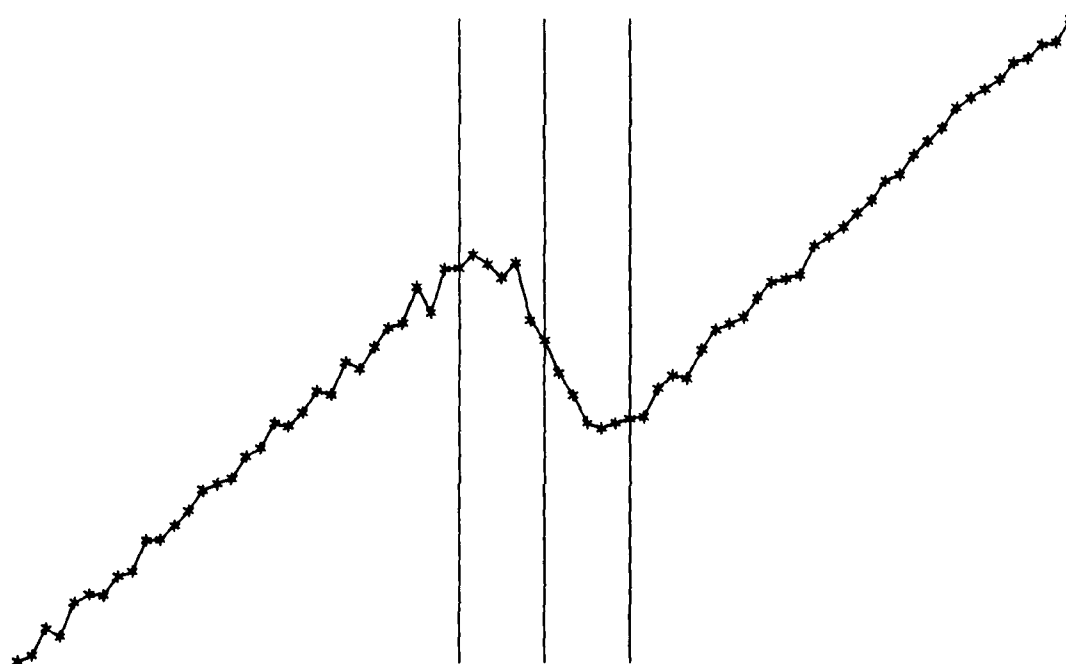
University of Utah



09:35:28

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point =  $-1.04E-17$   
Direction = 4.00  
F2 \* (0.9922, 1.002)

s =  $2.11E-04$   
h =  $1.26E-03$   
Number of Evaluations: 1353

This plot corresponding to figure

17

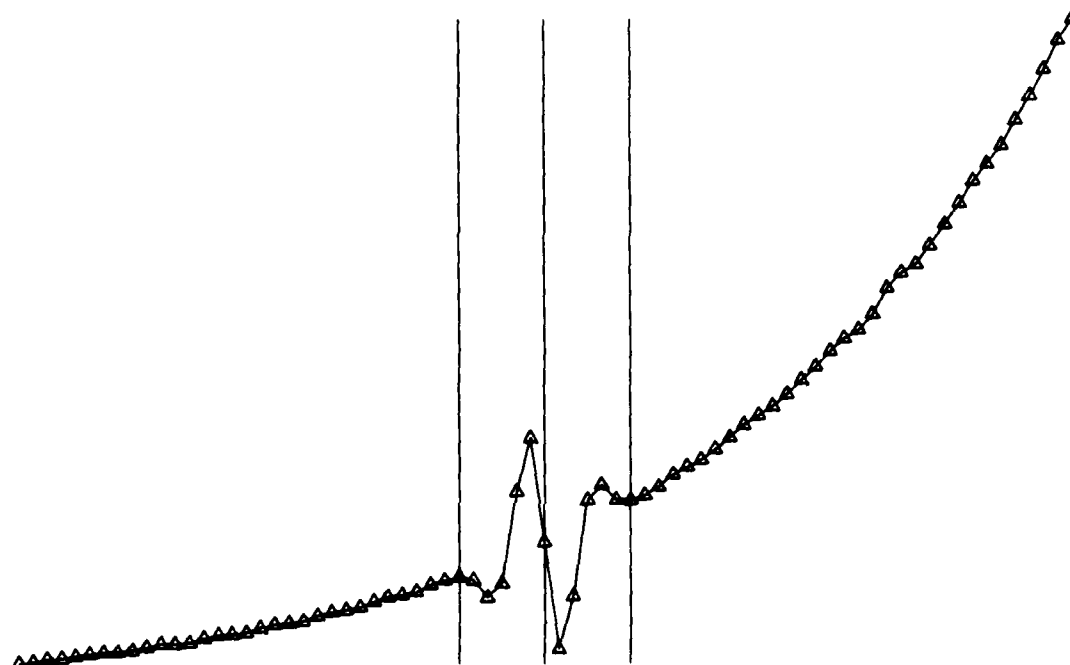
PLOT 79

University of Utah

09:36:05

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point =  $-1.04\text{E}-17$   
Direction = 4.00  
F6  $\Delta$  (0.2233, 4.395)

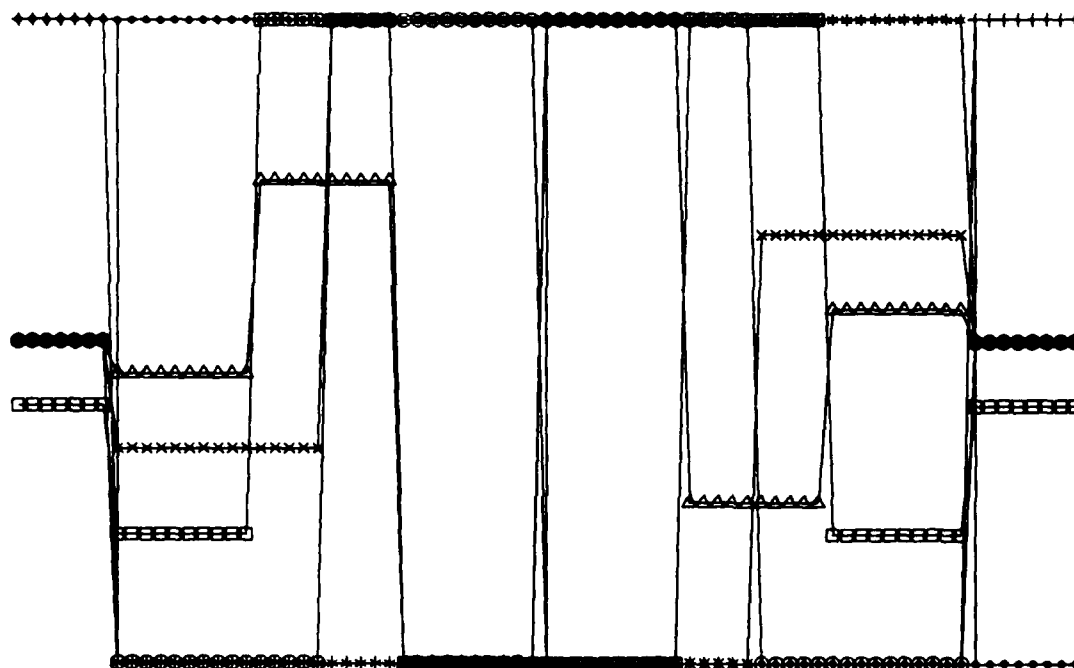
$s = 0.0400$   
 $h = 0.240$   
Number of Evaluations: 1440

This plot corresponding to figure  
18

P L O T 7 9

University of Utah

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F0 • (-1.0000, 9.977E-11)

F2 \* (-1.0000, 1.0000)

F4 × (-48.00, 48.00)

F6 △ (-7.290E+03, 7.290E+03)

s = 0.0333

h = 1.000

F1 + (-0.5000, 7.683E-11)

F3 ○ (-4.000, 4.000)

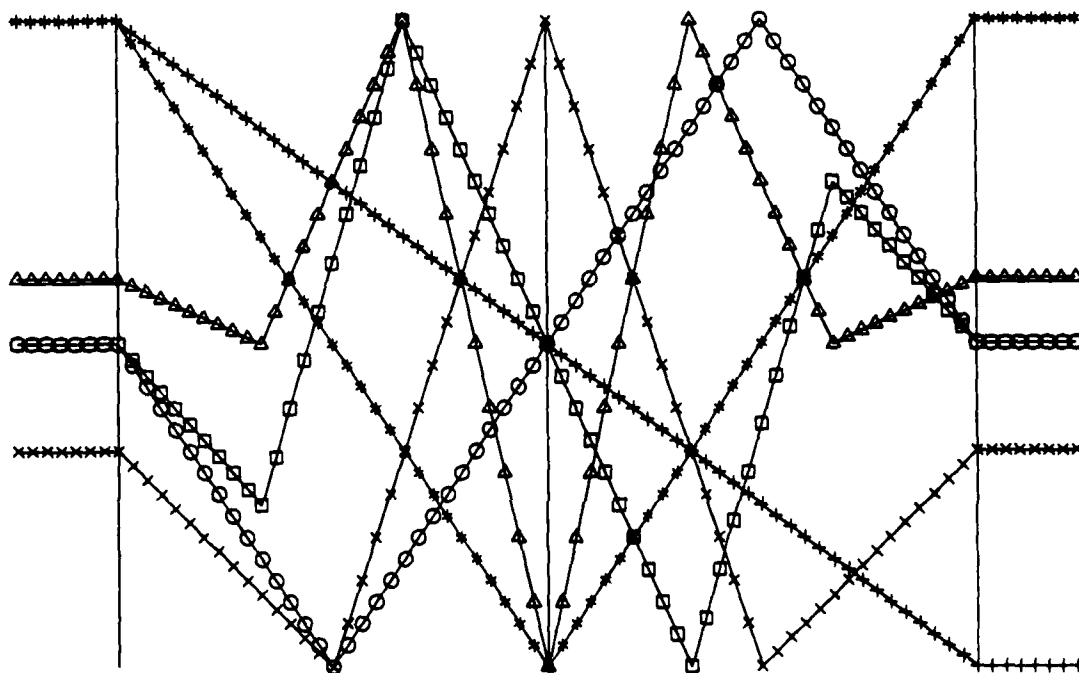
F5 □ (-2.430E+02, 3.645E+02)

Number of Evaluations: 1662

This plot corresponding to figure

19

## MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F1 + (-1.0000, 4.284E-11)

F3 o (-2.000, 2.000)

F5 □ (-81.00, 81.00)

Number of Evaluations: 1797

s = 0.0333

h = 1.000

F2 \* (-1.0000, 2.851E-10)

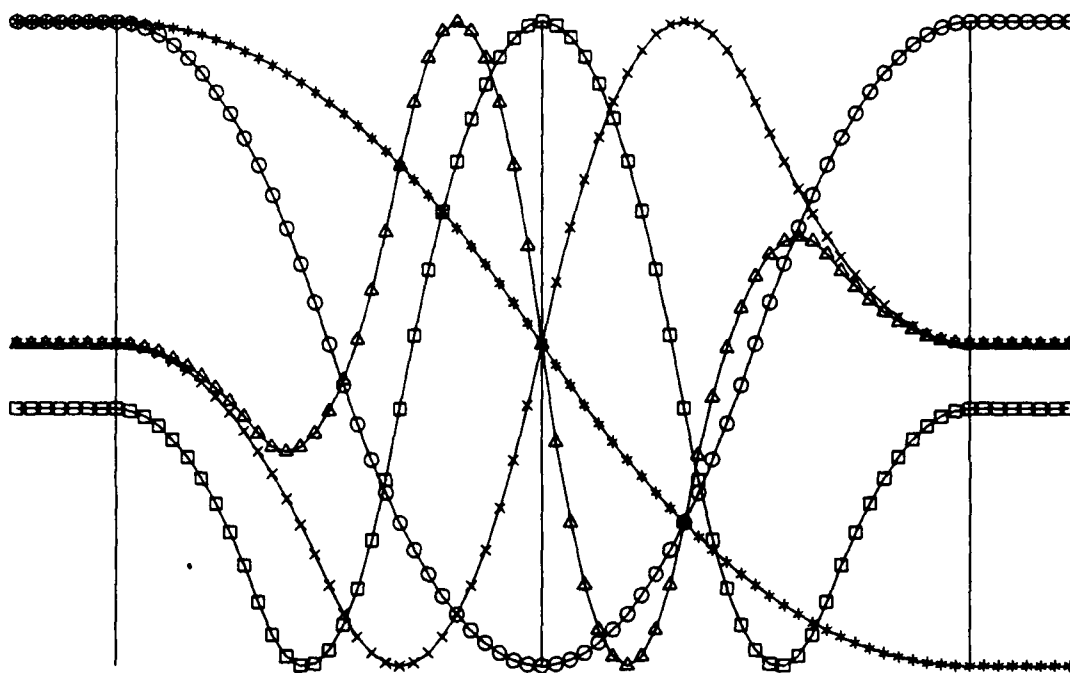
F4 x (-8.000, 16.00)

F6 △ (-1.458E+03, 9.720E+02)

This plot corresponding to figure

20

## MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F2 \* (-1.0000, 1.987E-10)

F4 x (-2.666, 2.666)

F6 Δ (-1.458E+02, 1.458E+02)

s = 0.0333

h = 1.000

F3 o (-1.0000, 1.736E-09)

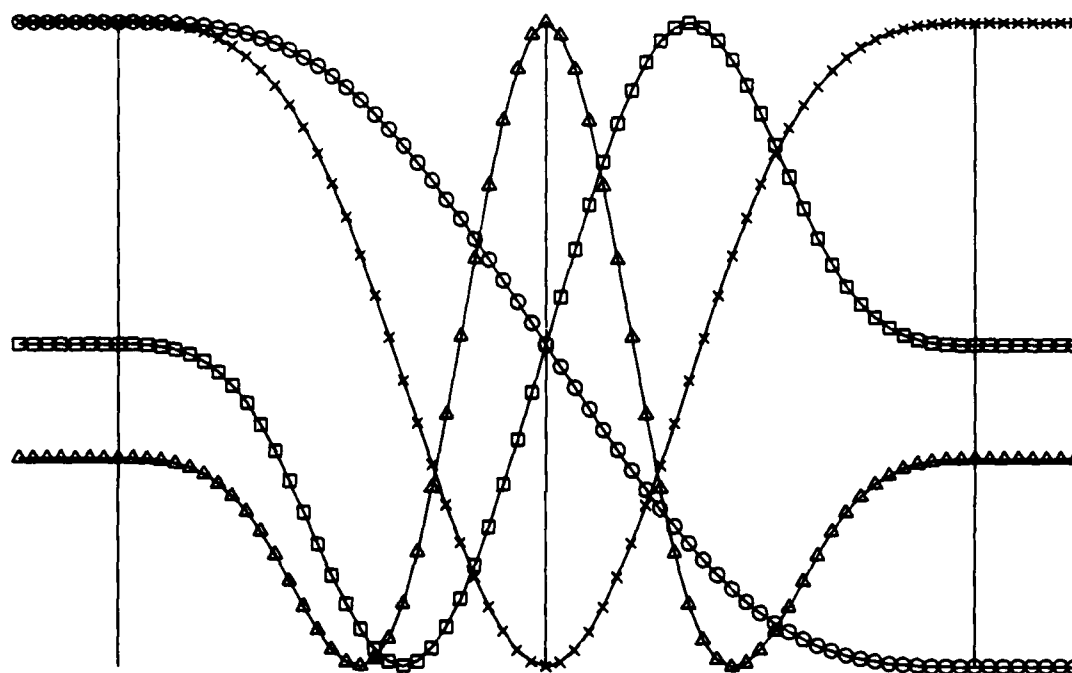
F5 □ (-8.977, 13.50)

Number of Evaluations: 1932

This plot corresponding to figure

21

## MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F3 ○ (-1.0000, 1.600E-09)

F5 □ (-3.000, 3.000)

Number of Evaluations: 2067

s = 0.0333

h = 1.000

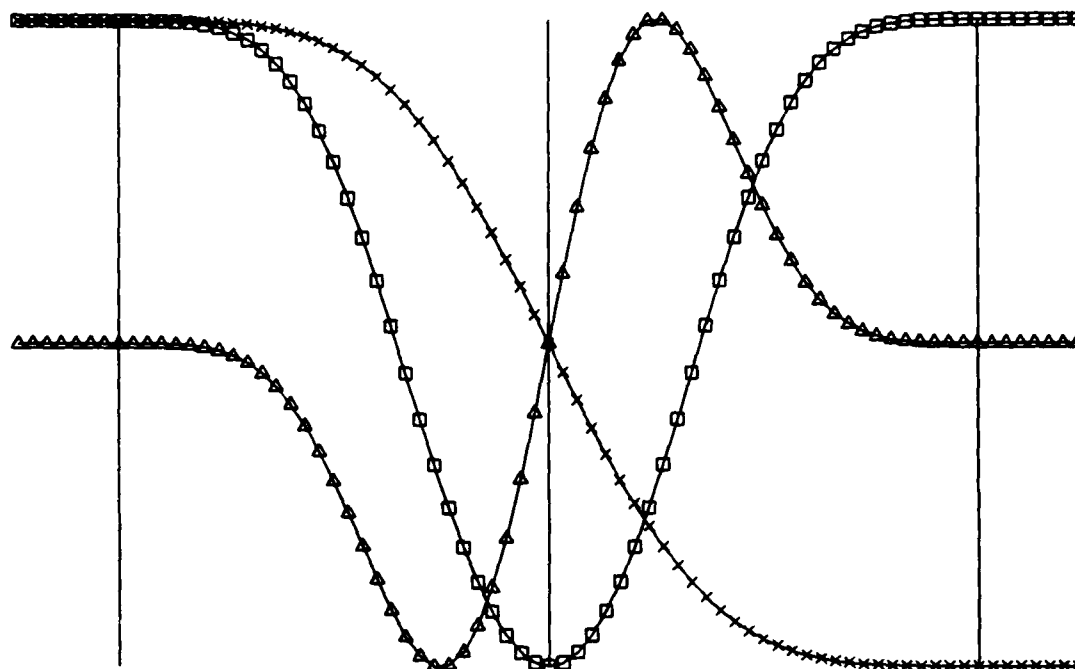
F4 × (-1.333, 1.139E-08)

F6 △ (-12.84, 27.00)

This plot corresponding to figure

22

## MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F4 x (-1.0000, 9.600E-09)

F6  $\Delta$  (-4.128, 4.128)

s = 0.0333

h = 1.000

F5  $\square$  (-1.375, 6.609E-08)

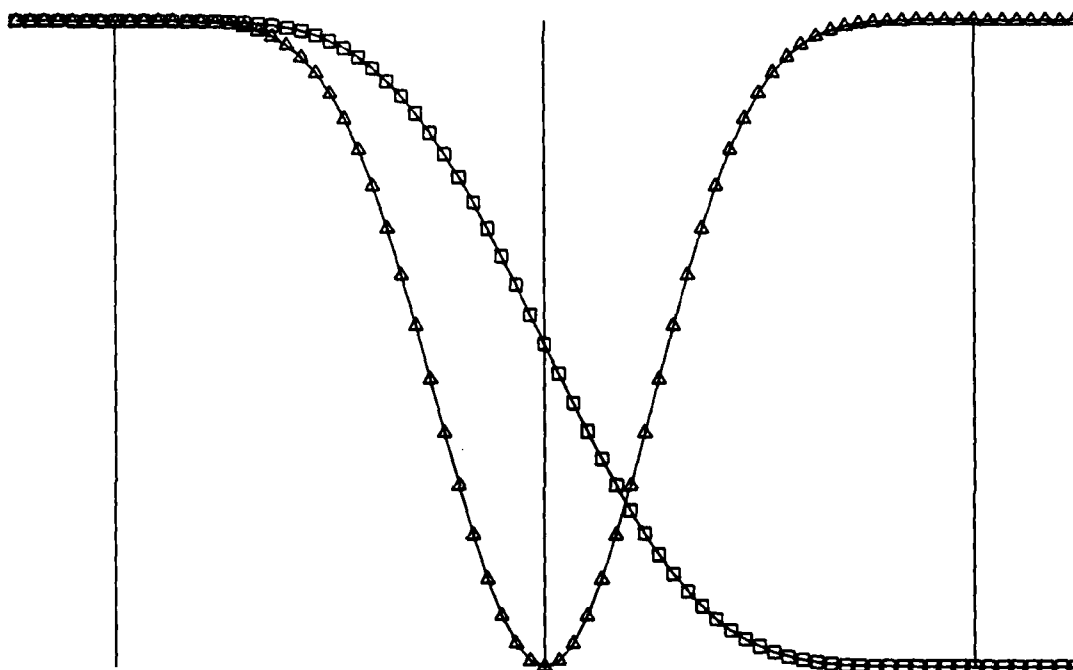
Number of Evaluations: 2202

This plot corresponding to figure  
23

09:48:06

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F5  $\square$  (-1.000, 5.093E-08)

Number of Evaluations: 2337

s = 0.0333

h = 1.000

F6  $\triangle$  (-1.650, 2.666E-06)

This plot corresponding to figure

24

PLOT 79

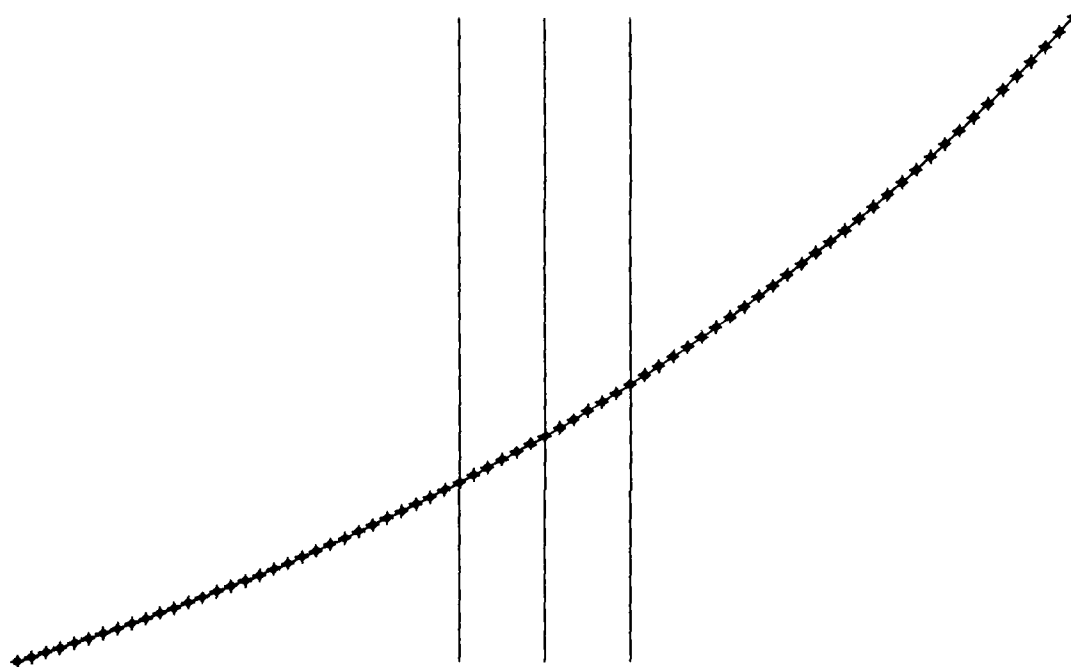
University of Utah



09:49:22

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F0 = (0.5397, 1.852)

Number of Evaluations: 2424

s = 0.0166

h = 0.1000

F1 = (0.5406, 1.855)

This plot corresponding to figure

25

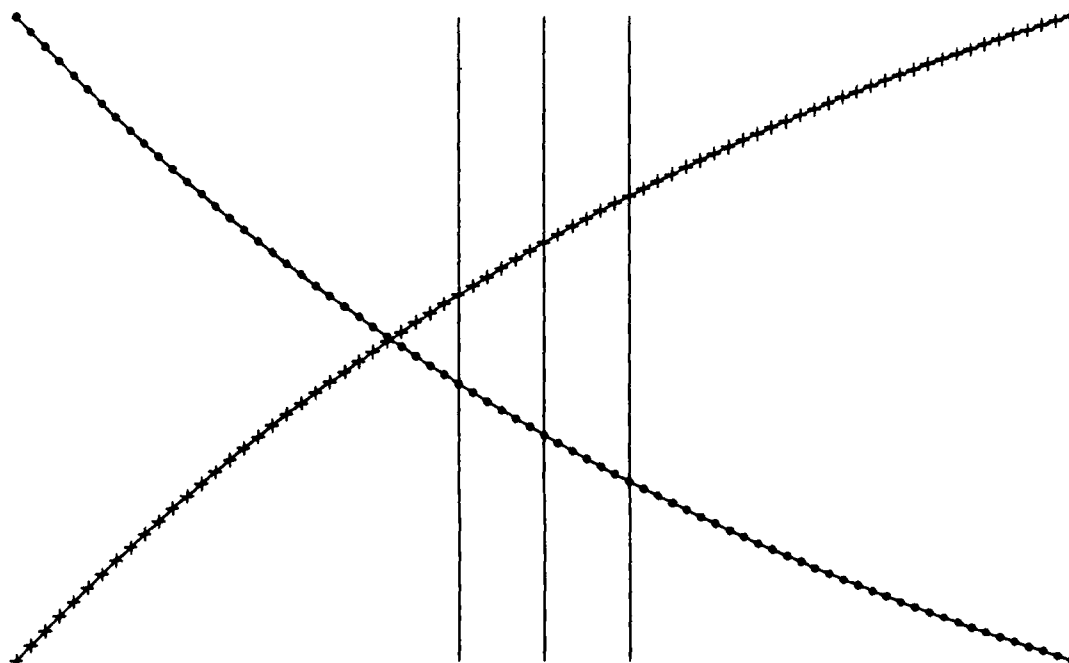
PLOT 79

University of Utah

09:50:10

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = -1.000

F0 = (0.5397, 1.852)

Number of Evaluations: 2424

s = 0.0166

h = 0.1000

F1 = (-1.855, -0.5406)

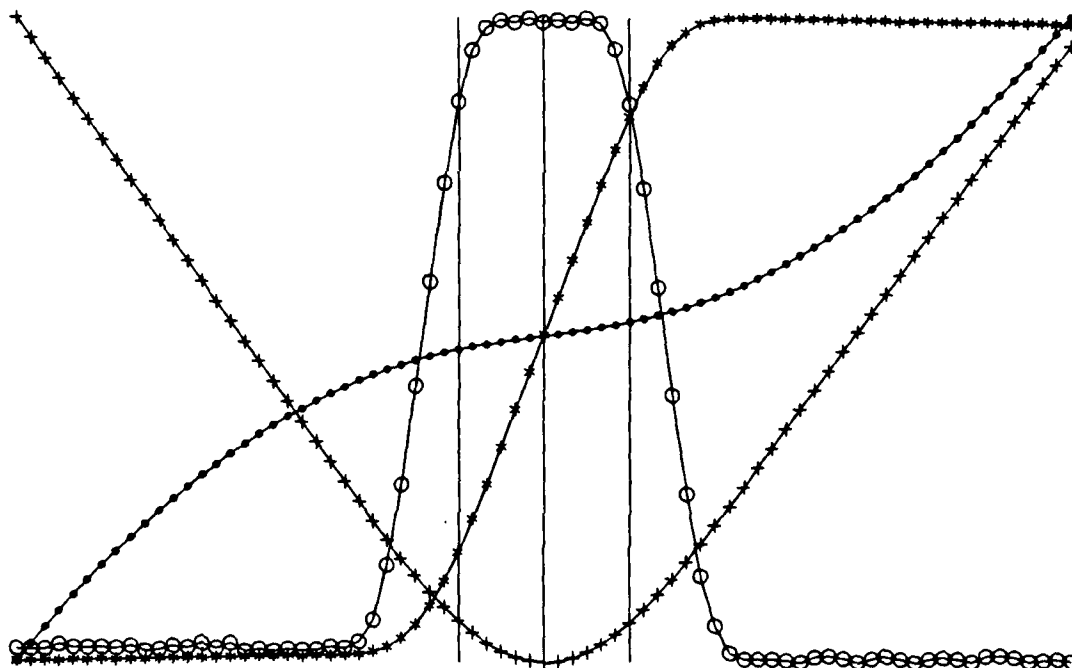
This plot corresponding to figure

26

P L O T 7 9

University of Utah

## MICROSCOPE: A Software System for Multivariate Analysis



CD: 1 Dir = (1.000, -1.000)

ch = 6.00E-03

Point = (0.0, 1.000)

s = 5.00E-04

Direction = (1.000, 0.0)

h = 3.00E-03

F0 • (-7.491E-04, 7.284E-04)

F1 + (0.01050, 0.08002)

F2 \* (-4.401, 4.183)

F3 o (-13.42, 1.002E+03)

Number of Evaluations: 2598

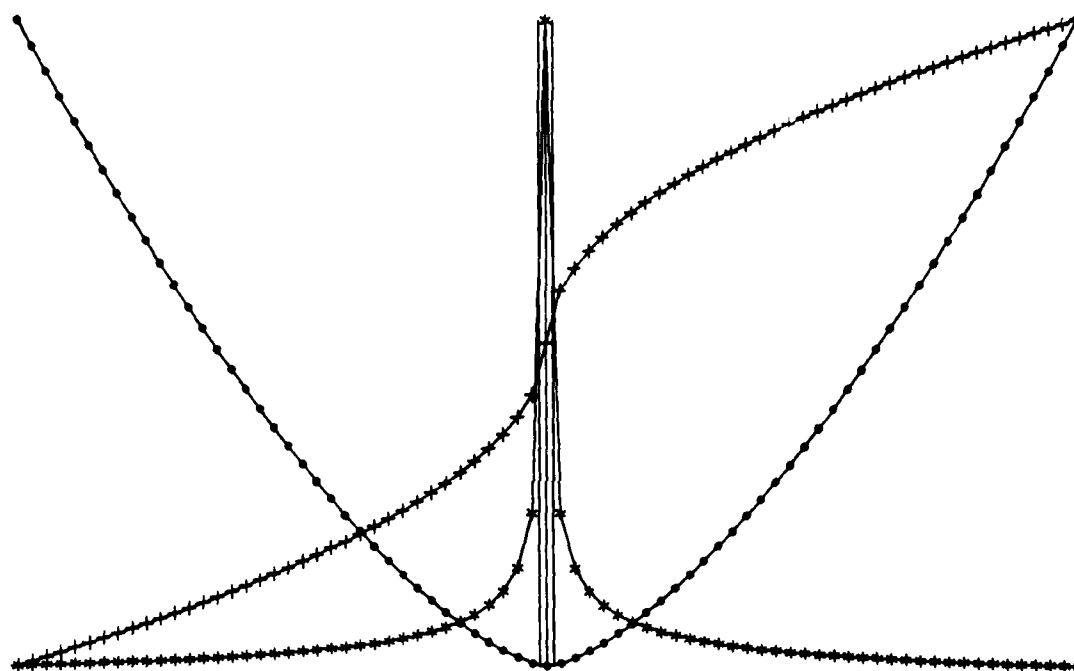
This plot corresponding to figure

27

09:55:29

27-APR-84

# MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F0 • (2.141E-11, 1.800E-03)

F2 \* (6.166, 1.414E+02)

s = 4.00E-04

h = 2.00E-04

F1 + (-0.1824, 0.1824)

Number of Evaluations: 2901

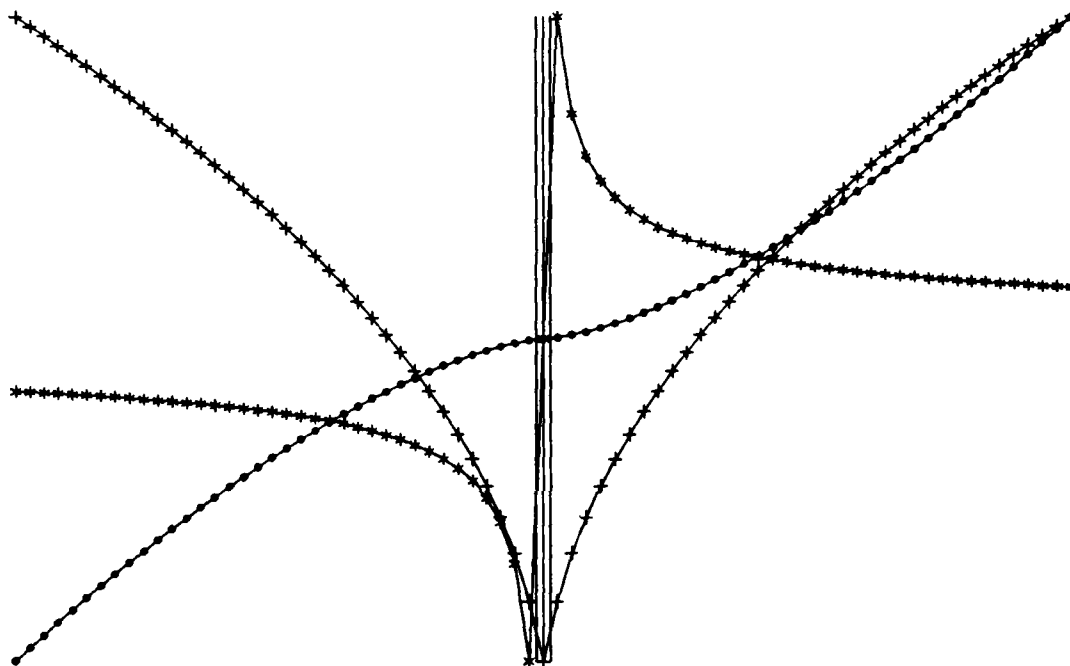
This plot corresponding to figure

28

P L O T 7 9

University of Utah

## MICROSCOPE: A Software System for Multivariate Analysis



Point = 0.0

Direction = 1.000

F0 • (-1.800E-03, 1.800E-03)

F2 \* (-38.13, 38.13)

s = 4.00E-04

h = 2.00E-04

F1 + (0.01414, 0.1824)

Number of Evaluations: 3052

This plot corresponding to figure

29

#### References

1. Abramowitz, M., and Stegun, I., 1968, Handbook of Mathematical Functions, Dover Publication
2. Alfeld, Peter, A Trivariate Clough-Tocher Scheme for Tetrahedral Data to appear in Computer Aided Geometric Design, An International Journal, North Holland
3. Beebe, Nelson H.F., 1980, DOCUMENT, A Portable Text Formatting Program, Department of Physics, University of Utah, Salt Lake City, Utah 84112
4. Beebe, Nelson H.F., 1979, A User's Guide to <PLOT79>, Department of Physics, University of Utah, Salt Lake City, Utah 84112
5. Barnhill, R.E., October 1983, A Survey of the Representation and Design of Surfaces, IEEE Computer Graphics and Applications, v. 3, Number 7, pp. 9-16
6. de Boor, C., 1978, A Practical Guide to Splines, Springer Verlag
7. Courant, R., and Hilbert, D., 1962, Methods of Mathematical Physics, v. 2, Wiley & Sons
8. Ryder, B.G., The PFORT Verifier, in Software Practice and Experience 4, pp. 369-377, Wiley & Sons
9. Schumaker, L., 1981, Spline Functions: Basic Theory, Wiley & Sons

#### Acknowledgements

MICROSCOPE was developed over several years and has evolved through several preliminary versions. This process was shaped by the work of the Computer Aided Geometric Design Group at the Mathematics Department of the University of Utah and has benefited from the interactions with Paul Arner, R.E. Barnhill, Nelson Beebe, Gerald Farin, Tom Jensen, Chip Petersen, Bruce Piper, Tracy Whelan, Andrew Worsey, and others.

This work was supported partly by the Department of Energy under Contract No. DE-AC02-82ER12046 A000 to the University of Utah, by the United States Army under Contract No. DAAG29-80C-0041, by two grants from the University of Utah Research Committee, and by a sabbatical leave of the first author from the University of Utah. A large part of the program development, and this documentation, were carried out at the Mathematics Research Center at the University of Wisconsin. Fred Sauer of MRC has been most helpful with all computer related problems.

# INDEX

- Abramowitz, 13, 116
- ACCENT, 51, 54-55, 108
- Alfeld, 1, 7, 30, 92, 116
- alpha, 83-85, 87, 89-90
- Barnhill, 8, 116-17
- Beebe, 6, 8, 69, 78, 116-17
- bivariate, 9, 16, 30
- BLSCRN, 77
- de Boor, 21, 116
- C1CROSS, 51, 65-66
- C2CROSS, 51, 65-66
- C3CROSS, 51, 65-66
- CBD, 104
- CBDSW, 104
- CBDU, 104
- CCHANNL, 51, 63, 67-68
- CDIRECTN, 51, 65-66
- CHVALUE, 51, 65-66
- Clough-Tocher, 30, 116
- CLSCRN, 77
- command, 3, 5-6, 17-18, 22, 33, 35-38, 41, 44-74, 78-79, 92-94, 101-2, 104, 107-11
- CORDER, 51, 65-66
- Courant, 45, 116
- critical, 9-10, 27, 85; see also set
- cross, 3, 14-16, 18-19, 24, 30, 44, 51-53, 58, 63-66, 68, 72-74, 104, 106, 109; see also derivative, direction
- CRT, 8, 10, 13, 15
- cuspid, 9, 49, 114
- CWINDOW, 51, 60-62
- D, 7, 14, 22-32, 38-43, 46-49, 51-53, 56-57, 59, 64, 67, 83, 85-86, 88, 111-14, 116
- d, 10-11, 16, 18, 51, 54-55, 59-60, 99-100
- D1CHDIR, 51, 59
- D2CHDIR, 51, 59
- D3CHDIR, 51, 59
- DCENTER, 51, 54-55, 72
- Delta, 10, 40, 45, 49-50, 83-85, 89-91
- DEN, 94
- derivative, 3, 5, 9-11, 14-16, 19, 21, 24-27, 30-32, 37-60, 64-66, 70, 73-74, 82, 84-86, 88-91, 98, 105-6, 109, 113-14
- DF, 105-6
- DGRAPH, 39, 51, 54-55, 108
- Dirac, 10, 45, 84
- direction, 3, 10-11, 14-16, 18, 22-32, 38-43, 46-49, 51-53, 55-60, 63-68, 72-74, 96, 99, 104, 106-8, 111-14; see also investigation
- display, 3-6, 8-11, 13-20, 22-35, 37-42, 44, 46, 49-51, 53-61, 63, 65-66, 69-72, 74, 76-78, 83-84, 93-98, 105-7, 109, 111-12, 115; see also function
- DIVIDE, 51, 60-62
- DMNSN, 38, 51, 54, 58, 65
- DOCUMENT, 6, 116
- DOUBLE, 34, 51, 60-62, 73, 104, 111
- DSCALE, 39, 51, 54-55, 72
- DXAXIS, 51, 54-55, 72
- e, 8-11, 15-19, 21-22, 27-28, 34, 37-39, 52, 54-55, 58-61, 63-64, 68-69, 71, 78, 82-83, 89, 92, 94, 98, 104-5, 107, 110-11, 113
- EGRAPH, 40, 51, 54-55, 108
- eps, 7, 83
- eta, 38, 42-43, 82-84, 89-90, 110-13
- examination, 3, 10, 14-16, 19, 23, 28, 31, 42, 50-53, 55, 58-59, 63-64, 66, 68, 71-72, 74, 82, 95-96, 99, 110, 114
- EXIT, 51, 62-63, 68
- External, 10
- F, 10-13, 34, 73, 78, 82, 110-11, 116
- FLIP, 51, 54-56
- Font, 77
- FORCE, 33, 51-52, 54, 61-64, 73,



# INDEX

- 92, 109
- front, 7, 10, 66, 112
- FS, 96, 100, 105, 108
- function, 2-5, 8-10, 13-17, 19-22, 24, 26-31, 34-36, 38, 40, 42-50, 55-56, 61-64, 66, 71, 73-74, 82-84, 89-93, 95-98, 100, 105, 110-14, 116
- Gateaux, 18
- GO, 18, 33, 38, 51-52, 54, 58, 61-64, 73, 80, 92, 94
- GRAPHIC, 34
- h, 5, 11-12, 14, 16, 22-32, 38-43, 46-52, 56-57, 60-61, 63-64, 66-67, 83, 85, 87-91, 95-97, 102, 104, 108, 113-14
- HALVE, 41, 51, 60-62
- Harris, 1, 92
- HELP, 34, 51, 58, 64-65, 79, 109
- HELPLC, 75, 79-81
- HELPUC, 75, 79, 81
- Hershey, 77; see also Font
- Hilbert, 45, 116
- HSUMRY, 51, 64-65, 79, 109
- ICBD, 104
- IDIRCTN, 51, 59
- IHVALUE, 51, 60-62
- IINTVL, 38, 51, 58-59
- investigation, 3, 7, 10, 14-19, 35, 51-53, 55, 58-60, 63-64, 66, 68, 72, 74, 95-98, 107-8, 112
- IPOINT, 51, 58
- ITYPE, 92-94
- k, 11, 13, 15-16, 45, 50-55, 57, 65-66, 70, 82-91
- LC, 75-76, 80-81
- LCBD, 104
- LCROSS, 51, 65-66
- LDEF, 100, 105
- LDRCTN, 51, 59
- LGO, 92-94, 107
- LINE, 34-35, 107
- LIST, 51, 64-65
- LOG, 51, 65, 105
- LPOINT, 51, 58-59
- LSCRN, 93, 107
- MANLC, 75, 80-81
- MANUAL, 75, 79-81
- MANUC, 75, 80-81
- MCRSCP, 3, 33-35, 58, 62-63, 68, 74, 76, 92-93, 101, 104
- MICROSCOPE, 1-4, 6-8, 10-11, 13-14, 17-23, 25, 32-45, 49, 52-54, 79, 84, 87, 92-110, 117
- Mode, 3, 35-37, 52-53, 57, 65, 68-69, 73, 94, 107, 110-11
- multivariate, 1-3, 8-9, 18-19, 44, 112
- MULTPLY, 52, 60-62
- NEWS, 52, 64-65
- nh, 96, 99, 101-4, 107
- NORMAL, 52, 59-60, 66, 108
- NUM, 94
- NUMRCL, 34-35, 109
- nw, 96-97, 99, 101-4
- OUTPUT, 34-35, 52, 65, 107
- P, 10-13, 16, 45, 51-53, 58, 73, 99-100
- P1CHPNT, 52, 58
- P2CHPNT, 52, 58
- P3CHPNT, 52, 58
- package, 3-4, 6-9, 13, 17, 20, 35, 38, 64, 68-69, 75-79, 92-94, 104, 110-14
- PAUSE, 52, 67-68
- PCURSR, 77
- PEN07, 70, 75, 78
- PEN18, 70, 75, 78
- PFORT, 8, 116; see also Verifier, Portable, Portability
- PLCHRS, 77
- PLOT, 3, 34-35, 37, 52, 69-72, 74, 77-78, 109-10
- PLOT79, 3-4, 8, 35, 52, 54, 69-72, 77-78, 80, 115-16
- PLTDMY, 75, 78, 80-81
- PLTLC, 75, 77-78, 80-81
- PLTUC, 75, 77-78, 81
- point, 3, 6, 10-11, 13-16, 18-19, 22-32, 35, 37-43, 45-49, 51-59, 62-64, 66-68, 70-74, 77-78, 82-85, 95-100, 105-8, 111-14; see

# INDEX

- also
  - examination
- pole, 9, 113-14
- Portability, 6, 33, 75
- Portable, 2, 8, 33, 75-76, 79, 116
- POS, 75-76, 80-81
- precision, 3, 6, 28-29, 36, 73, 104
- primitive, 8, 10, 21-22, 27, 44;
  - see also function
- PROMPT, 34-35
- PS, 99-100
- psi, 85, 89-90
- QUIT, 52, 62-63
- random, 7, 10, 23-24, 45, 83, 112
- RCROSS, 52, 65-66, 68
- RDIRCTN, 52, 59, 68
- RESTART, 34, 52, 62-63, 111
- ROPPNT, 100, 107-8
- ROPUDI, 100, 108
- ROTATE, 52, 59
- round-off, 3-5, 10, 13, 22-24, 28-32, 35, 42, 44-45, 60-61, 82-90, 94, 112
- RPOINT, 52, 58-59, 68
- RSCREEN, 33, 52, 54, 58, 94
- RWIND, 52, 67-68
- Ryder, 8, 116
- s, 11, 14, 16, 22-32, 38-43, 46-50, 56-57, 60, 67, 84-85, 95, 101-2, 110, 113-14
- scale, 15, 39, 41, 51, 54-55, 63-64, 72, 74, 106, 108
- Schumaker, 21, 116
- SCREEN, 75-77, 80-81, 93, 107, 109
- screen, 6, 8, 10, 13, 17-18, 33-39, 51-54, 57-58, 61-65, 69, 72, 76-78, 80-81, 92, 94, 101, 105-8, 110
- SCROLL, 75-77, 81
- scroll, 33, 94
- set, 7, 9-10, 14-15, 17-18, 27-28, 34, 42, 59, 63-66, 75, 84, 92, 94, 99, 102, 104, 107-8
- SETDF, 52, 62-63, 111
- SETPEN, 70-71
- SETPN, 71
- SHIFT, 39, 52, 58-59
- spline, 21, 111, 116
- ss, 95-99, 101-2, 104
- Stegun, 13, 116
- STORE, 52, 62-63, 111
- STPEN, 71
- SUBUSR, 35, 73-74, 78, 110
- SUPPRT, 75-76, 80-81
- surface, 9, 19
- sw, 95-98, 100-102, 108
- tangential, 11, 14-16, 18-19, 44-45, 54-55, 66, 70-71, 73-74, 82, 92, 97, 109;
  - see also derivative
- TCENTER, 52, 54, 57
- terminal, 8, 10, 15, 17, 19, 33-34, 54, 68, 77-78, 101, 105, 107
- test, 4-9, 38, 44, 64, 78, 110-14;
  - see also package
- TNOTE, 52, 65
- trial, 8-10, 13-16, 19, 34-36, 38, 40, 42, 44, 49, 60-63, 66, 73-74, 82, 89, 110-12, 114;
  - see also function
- trivariate, 9, 116
- TYPE, 37, 42, 52, 54, 57
- UC, 75-76, 81
- UNDO, 52, 62, 64, 107
- USER, 37-38, 52, 62-63, 73-74, 78, 104, 109-11
- USRDMY, 75, 78
- USRLC, 75, 78, 80-81, 110
- USRUC, 75, 78, 81, 110
- Verifier, 8, 116
- w, 16, 49-50, 60, 63-64, 85
- WAIT, 52, 62-63
- WIDTH, 34-35, 107
- width, 5, 15-16, 19, 35, 41, 49-52, 60-61, 63-64, 74, 85, 95-98, 101-4, 108, 113
- window, 3, 5, 16, 19, 41-42, 46, 49-52, 60-61, 63-64, 66, 74, 85, 89, 98, 101, 108, 113
- XS, 96, 99-100, 105
- ZOOM, 52, 60-62

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER  2701	2. GOVT ACCESSION NO.  AD-414472C	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  MICROSCOPE: A SOFTWARE SYSTEM FOR MULTIVARIATE ANALYSIS		5. TYPE OF REPORT & PERIOD COVERED Summary Report - no specific reporting period
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)  Peter Alfeld and Bill Harris		8. CONTRACT OR GRANT NUMBER(s)  DAAG29-80-C-0041 DE-AC02-82ER12046 A000
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematics Research Center, University of 610 Walnut Street Madison, Wisconsin 53706		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit Number 3 - Numerical Analysis and Scientific Computing
11. CONTROLLING OFFICE NAME AND ADDRESS  See Item 18 below.		12. REPORT DATE June 1984
		13. NUMBER OF PAGES 120
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)  UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES U. S. Army Research Office P. O. Box 12211 Research Triangle Park North Carolina 27709 Department of Energy Washington, D. C. 20545		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Interpolation Approximation Software Several Variables Computer Aided Geometric Design		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This manual describes <u>MICROSCOPE</u> : a portable FORTRAN software system for the analysis of multivariate functions. Given an interpolation or approximation scheme, it allows the following questions, among others, to be answered: Does the scheme interpolate? How often is it differentiable? What functions does it reproduce exactly? If the scheme is polynomial, what is its polynomial degree? Where is the smoothness of a function reduced? Where are the bugs in a FORTRAN implementation?		

**DATE**  
**ILME**